INTERNATIONAL STANDARD

ISO/IEC 8327-1

Second edition
1996-09-15

AMENDMENT 1
1998-10-01

Information technology — Open Systems Interconnection — Connection-oriented Session protocol: Protocol specification

AMENDMENT 1: Efficiency enhancements

Technologies de l'information — Interconnexion de systèmes ouverts (OSI) — Protocole de session en mode connexion: Spécification du protocole

AMENDEMENT 1: Améliorations d'efficacité

Reference number
ISO/IEC 8327-1:1996/Amd.1:1998(E)

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Amendment 1 to ISO/IEC 8327-1:1996 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 33, *Distributed application services*, in collaboration with ITU-T. The identical text is published as ITU-T Rec. X.225/Amd.1.

INTERNATIONAL STANDARD

ITU-T RECOMMENDATION

INFORMATION TECHNOLOGY – OPEN SYSTEMS INTERCONNECTION – CONNECTION-ORIENTED SESSION PROTOCOL: PROTOCOL SPECIFICATION

AMENDMENT 1
Efficiency enhancements

1) Subclause 2.1

Add the following reference by numerical order:

– ITU-T Recommendation X.215 (1995)/Amd.1 (1997) | ISO/IEC 8326:1996/Amd.1:1997, *Information technology – Open Systems Interconnection – Session service definition – Amendment 1: Efficiency enhancements.*

2) Subclause 2.3

Add the following reference after Recommendation T.62:

– ITU-T Recommendation X.215 (1995)/Addendum 1 (1995), *Service definition for Session Layer efficiency enhancement.*

3) Subclause 3.4

Add the following definitions:

3.4.a long-form SPDU: An SPDU that has the long-form structure defined in 8.2.

3.4.b null-encoding protocol option: An option of the session protocol, negotiated during connection establishment, that permits a data transfer phase with zero session protocol control information and without the ability to signal the orderly release of the session-connection.

3.4.c parameter indication: A field in the low-order bits of the first octet of a short-form SPDU (the high-order bits will contain the SPDU identifier).

3.4.d short-connect protocol option: An option of the session protocol that permits an efficient negotiation, during connection establishment, of the fast associate mechanism (of which the null-encoding protocol defined in ITU-T Rec. X.225 *bis* is a special case) by defining more compact encodings for the connection establishment SPDUs than those defined in ITU-T Rec. X.225 | ISO/IEC 8327-1.

3.4.e short-encoding protocol option: An option of the session protocol that permits the use of smaller protocol control information of some of the more commonly occurring Session SPDUs in the data transfer and release phases.

3.4.f short-form SPDU: An SPDU that has the short-form structure defined in 8.5. All short-form SPDUs have names that begin with the word SHORT and abbreviations beginning with the letter S.

4) Subclause 4.2

Add at the end of the abbreviations list:

SI&P SPDU Identifier (for short-form SPDUs) and Parameter indication

5) Subclause 5.2

In Table 1, referenced in this subclause, add the following items to the cell identified by Session Connection, Associated SPDUs:

Service	Primitives	Associated SPDUs
Session Connection	S-CONNECT request S-CONNECT indication	CONNECT SPDU or SHORT CONNECT SPDU
	S-CONNECT (accept) response	CONNECT SPDU or SHORT CONNECT SPDU ACCEPT SPDU, SHORT ACCEPT SPDU or
	S-CONNECT (accept) confirm	SHORT ACCEPT CONTINUE SPDU ACCEPT SPDU, SHORT ACCEPT SPDU or
	S-CONNECT (reject) response	SHORT ACCEPT CONTINUE SPDU REFUSE SPDU, SHORT REFUSE SPDU or
	S-CONNECT (reject) confirm	SHORT REFUSE CONTINUE SPDU REFUSE SPDU, SHORT REFUSE SPDU or SHORT REFUSE CONTINUE SPDU
Normal Data Transfer	S-DATA request	DATA TRANSFER SPDU or SHORT DATA TRANSFER SPDU or NULL SPDU
	S-DATA indication	DATA TRANSFER SPDU or SHORT DATA TRANSFER SPDU or NULL SPDU
Orderly Release	S-RELEASE request	FINISH SPDU or SHORT FINISH SPDU
	S-RELEASE indication	FINISH SPDU or SHORT FINISH SPDU
	S-RELEASE(accept) response	DISCONNECT SPDU or SHORT DISCONNECT SPDU
	S-RELEASE(accept)indication	DISCONNECT SPDU or SHORT DISCONNECT SPDU
	S-RELEASE(reject) response	NOT FINISHED SPDU
	S-RELEASE(reject)indication	NOT FINISHED SPDU

6) Subclause 5.4.2

Add at the end of the list and before the Note, the following new items:

h) to negotiate the null-encoding protocol option (see 5.8.7);

i) to negotiate the upper layer context specification.

7) Subclause 5.6

Add a new subclause as follows:

5.6.2 *bis* No orderly release functional unit

This functional unit removes the orderly release function from the kernel functional unit.

8) Subclause 5.6.10

Modify Table 3 referenced in this subclause:

Functional Unit	SPDU code	SPDU name	Reference
Kernel	CN	CONNECT (see Note 1)	7.1
	OA	OVERFLOW ACCEPT (see Note 2)	7.2
	CDO	CONNECT DATA OVERFLOW (see Note 2)	7.3
	AC	ACCEPT (see Note 1)	7.4
	RF	REFUSE (see Note 1)	7.5
	FN	FINISH (see Note 10)	7.6
	DN	DISCONNECT (see Note 10)	7.7
	AB	ABORT (see Note 11)	7.9
	AA	ABORT ACCEPT (see Note 3)	7.10
	DT	DATA TRANSFER (see Note 11)	7.11
	PR	PREPARE (see Note 7)	7.26
	SCN	SHORT CONNECT (see Note 14)	7.38
	SAC	SHORT ACCEPT (see Note 14)	7.39
	SRF	SHORT REFUSE (see Note 14)	7.42
	NL	NULL (see Note 13)	7.49
	SCNC	SHORT CONNECT CONTINUE (see Note 14)	7.40
	SACC	SHORT ACCEPT CONTINUE (see Note 14)	7.41
	SRFC	SHORT REFUSE CONTINUE (see Note 14)	7.43
	SFN	SHORT FINISH (see Note 12)	7.44
	SDN	SHORT DISCONNECT (see Note 12)	7.45
	SDT	SHORT DATA TRANSFER (see Note 12)	7.46
	SAB	SHORT ABORT (see Note 12)	7.47
No orderly release		No additional associated SPDUs	

After Note 9 of this table, add the following Notes:

10 Not used if the no orderly release functional unit is selected.

11 Not used if the null-encoding protocol option is selected.

12 Used only if the short-encoding protocol option is supported.

13 Used only if the null-encoding protocol option is supported.

14 Used only if the short-connect protocol option is selected.

9) Subclause 5.8

Add the new subclauses (at the end of 5.8.6):

5.8.7 Negotiation of short-encoding

Each SPM indicates whether it wishes to use the short-encoding option on the connection. The protocol option is selected only if both SPMs propose use of the option. If the option is selected, the SHORT DATA TRANSFER, SHORT FINISH, SHORT DISCONNECT and SHORT ABORT SPDUs may be used on the connection.

The use of the other short-form SPDUs (SHORT CONNECT, SHORT ACCEPT, SHORT CONNECT CONTINUE, SHORT REFUSE CONTINUE, SHORT ACCEPT CONTINUE and SHORT REFUSE SPDUs) is not affected by the short-encoding protocol option.

5.8.8 Negotiation using short-connect mechanism and Upper-layer context specification

An initiating SPM receiving an S-CONNECT request that includes a Session-user-summary parameter may use the short-connect mechanism. Conceptually, the SPM creates the CONNECT SPDU that would be used to establish the connection. The SPM then uses an Upper-layer context specification to summarize the parameters of this CONNECT SPDU, including the semantic content of the User-data as represented in the User-summary parameter of the S-CONNECT request.

The Upper-layer context specification is identified in the session protocol by either a global-form identifier (an ASN.1 Object Identifier) or a restricted-form identifier (a 16-bit quantity that is unambiguous within the scope of some community of interconnecting systems).

The Upper-layer context specification may or may not define parameters that reflect values of the parameters of the CONNECT SPDU, or, via the Session-user-summary parameter, values contained within the SS-userdata of the S-CONNECT request. The Upper-layer context specification will define that each parameter is either:

a) immediate: always sent with the identifier; or

b) compressible: a compressed form can be sent with the identifier, and the original form sent on a subsequent SPDU if the receiver is unable to reverse the compression.

NOTE 1 – The Upper-layer context specification will define the compression algorithm.

The SPM will send the identifier for the Upper-layer context specification and any immediate or compressed parameters in the Connection summary parameter of a SHORT CONNECT SPDU. This SPDU may be sent on the User-data of a T-CONNECT request or on the normal transport flow, using T-DATA.

The SHORT CONNECT SPDU shall only be sent on the T-CONNECT request if the size limitations of the Transport layer permit. How the SPM is made aware of these limits is a local matter.

One of the following will then occur:

a) The responding SPM is able to reference the Upper-layer context specification and expand any compressed parameters to their original form, and is thus able to reconstruct the CONNECT SPDU that would have been sent. If acceptable to the SPM, an S-CONNECT indication is issued to the SS-user, with no User-data parameter but with the User Summary parameter representing the semantic content of the User-data that would have been sent.

   If the SS-user replies with an S-CONNECT (accept) response, the SPM uses the Upper-layer context specification identified on the SHORT CONNECT SPDU to determine the identification of the Summary-response. Again this may have immediate and compressed parameters. The identification and any immediate or compressed parameters are sent in the Summary-response parameter of a SHORT ACCEPT SPDU, which also indicates that connection establishment is complete.

   The receipt of the SHORT ACCEPT SPDU at the initiator completes the connection establishment.

   NOTE 2 – It is expected that the Upper-layer context identifier together with the short connect PCI will be designed to fit within the size limitation of the T-CONNECT User-data.

b) The responding SPM is able to reference the Upper-layer context specification but there are compressed parameters that the SPM cannot expand to their original form. The SPM asks for the uncompressed forms to be sent, by sending a SHORT ACCEPT SPDU indicating an incomplete connection establishment.

   The initiating SPM, on receiving the SHORT ACCEPT SPDU, sends a SHORT CONNECT CONTINUE SPDU containing the uncompressed forms of the parameters. The responding SPM can now, if the received SPDUs are acceptable, issue an S-CONNECT indication with the semantic content of the missing User-data represented by the User Summary parameter.

   If the SS-user replies with an S-CONNECT (accept) response, the SPM uses the Upper-layer context specification identified on the SHORT CONNECT SPDU to determine the identification of the Summary-response and sends this, with any parameters in their uncompressed form on a SHORT ACCEPT CONTINUE SPDU.

c) The responding SPM is unable to reference the Upper-layer context specification – the identifier is not recognized – and the SHORT CONNECT SPDU was received on an established connection. The SPM replies with SHORT REFUSE SPDU indicating the connection summary is unknown.

   On receiving the SHORT REFUSE SPDU, the initiating SPM switches to using the long-form SPDUs for connection establishment, sending the original CONNECT SPDU including any User-data.

d) The responding SPM is unable to reference the Upper-layer context specification – the identifier is not recognized – and the SHORT CONNECT SPDU was received on a T-CONNECT indication. The SPM ignores it, and just completes the establishment of the transport connection.

   On receiving the T-CONNECT confirm with no SPDU in the User-data, the initiating SPM switches to using the long-form SPDUs for connection establishment, sending the original CONNECT SPDU.

e) The responding SPM does not support the SHORT CONNECT SPDU and the SHORT CONNECT SPDU was received on an established connection. The SPM will perceive this as a protocol error and release the transport connection.

f) The responding SPM does not support the SHORT CONNECT SPDU and the SHORT CONNECT SPDU was received on a T-CONNECT indication. The SPM ignores it, and just completes the establishment of the transport connection.

On receiving the T-CONNECT confirm with no SPDU in the User-data, the initiating SPM switches to using the long-form SPDUs for connection establishment, sending the original CONNECT SPDU.

5.8.9 Negotiation of null-encoding protocol option

The use of the null-encoding protocol option is negotiated between the peer SPMs at session-connection establishment. It shall only be offered by the initiating SPM if the initiating Session user has requested, and the initiating SPM supports, the no-orderly-release functional unit. In addition, it can only be offered by the initiating SPM if no session layer addressing information is required to be conveyed

The null-encoding protocol option is selected for use on the session connection by the responding SPM. It shall only select the use of the option if:

a) the null-encoding protocol option was offered by the initiating SPM;

b) the responding SPM has selected the kernel, full-duplex and no-orderly-release functional units, and no other functional units.

5.8.10 Negotiation using short-connect protocol option with no upper-layer context identifier

The SPMs may use the short-connect protocol option to establish a session connection using the null-encoding option. The short-connect protocol option, as applied to connection establishment, uses the SHORT CONNECT SPDU, SHORT ACCEPT SPDU, SHORT ACCEPT CONTINUE SPDU and (if unsuccessful) the SHORT REFUSE SPDU and SHORT REFUSE CONTINUE SPDU.

The short-connect protocol option can only be used by the initiating SPM if, on the S-CONNECT request primitive:

a) the Session Connection Identifier parameter is absent;

b) in the Calling Session Address and Called Session Address, the session selector is absent; and

c) the Session Requirements parameter specifies the full duplex and no-orderly-release functional unit and no others.

The responding SPM can only issue a SHORT ACCEPT SPDU if, on the S-CONNECT response primitive:

a) the Session Connection Identifier parameter is absent;

b) in the Responding Session Address, the session selector is absent;

c) the Result is "accepted"; and

d) the Session Requirements parameter specifies the full duplex and no-orderly-release functional unit and no others.

The SHORT CONNECT SPDU, SHORT ACCEPT SPDU and SHORT REFUSE SPDUs can be transferred as User-data on the Transport-layer T-CONNECT primitives or as User-data on T-DATA primitives, if the Transport connection is already established. The mapping to the User-data of the T-CONNECT primitives is only possible if the complete SPDUs, including any User-data, meet any size restrictions of the T-CONNECT User-data. Otherwise procedures are applied to send the SPDUs using the T-DATA primitives.

10) Subclause 6.1.4

Add the following phrase to the antepenultimate paragraph of this subclause:

Only the initiator of the transport connection is permitted to issue the CONNECT SPDU or the SHORT CONNECT SPDU.

Replace the last paragraph of 6.1.4 with the following new paragraphs:

The TS-user data parameter in the T-CONNECT request and indication is used for the SHORT CONNECT SPDU. The TS-user data parameter in the T-CONNECT response and confirm is used for the SHORT ACCEPT SPDU and SHORT REFUSE SPDU if they fit, or for the SHORT ACCEPT CONTINUE SPDU and the SHORT REFUSE CONTINUE SPDU otherwise. When a T-CONNECT request is issued, the TS-user data parameter shall either contain a SHORT CONNECT SPDU or shall be empty. When a T-CONNECT response is issued, the TS-user data parameter shall be empty, unless the T-CONNECT indication contained a SHORT CONNECT SPDU, in which case it shall contain a SHORT ACCEPT SPDU, SHORT REFUSE SPDU, SHORT ACCEPT CONTINUE SPDU or SHORT REFUSE CONTINUE SPDU.

If the responding session implementation does not support the short-connect protocol option, it shall ignore the TS-user data parameter on the T-CONNECT indication and confirm.

11) Subclause 6.3.3

Add the following items at the end of the list of SPDUs:

- NULL SPDU (see 7.49);
- SHORT CONNECT (see 7.38);
- SHORT ACCEPT (see 7.39);
- SHORT CONNECT CONTINUE (see 7.40);
- SHORT ACCEPT CONTINUE (see 7.41);
- SHORT REFUSE (see 7.42);
- SHORT REFUSE CONTINUE (see 7.43);
- SHORT FINISH (see 7.44);
- SHORT DISCONNECT (see 7.45);
- SHORT DATA TRANSFER (see 7.46);
- SHORT ABORT (see 7.47).

Consequently, change the period of the last SPDU of the list to semi-colon.

12) Subclause 6.3.5

Change the first sentence as follows:

Segmenting of SSDUs takes place under the following circumstances, provided the null-encoding option has not been selected.

13) Subclause 6.3.7

Add the following to the column of Category 1 SPDUs *in Table 6:*

NULL SPDU
SHORT CONNECT
SHORT ACCEPT
SHORT CONNECT CONTINUE
SHORT ACCEPT CONTINUE
SHORT REFUSE
SHORT REFUSE CONTINUE
SHORT FINISH
SHORT DISCONNECT
SHORT DATA TRANSFER
SHORT ABORT

14) Subclause 6.4.4

Replace item b) with:

b) ABORT SPDUs are sent on the normal transport flow unless the null-encoding option is selected, in which case the ABORT SPDUs are not sent.

15) Subclause 6.6.4

Change the title of this subclause as follows:

6.6.4 Description (when null-encoding option is not selected)

Add the following new subclause after 6.6.4:

6.6.5 Description (when null-encoding option is selected)

When the null-encoding option is selected, the session connection is terminated by disconnection of the supporting transport connection.

16) Subclause 7.1

Add at the end of the paragraph:

The CONNECT SPDU is transmitted when the initiating SPM has chosen not to use a SHORT CONNECT SPDU, or after a SHORT CONNECT SPDU was transmitted on a T-CONNECT request and no SPDU was received on the T-CONNECT confirm, or after a SHORT REFUSE SPDU has been received with reason code value indicating "unknown connection summary".

17) Subclause 7.1.1

Replace item b) 1) with:

b) A Connect/Accept Item parameter group containing:

1) A Protocol Options parameter which enables the initiator to indicate its ability to receive extended concatenated SPDUs, to use the null encoding option and its ability to receive the following short-form SPDUs – SHORT DATA TRANSFER, SHORT FINISH, SHORT DISCONNECT and SHORT ABORT SPDU.

The initiator is not able to use the null encoding option unless the no-orderly-release functional unit was proposed by the calling SS-user.

18) Subclause 7.1.2

Replace the first two sentences by:

The sending of a CONNECT SPDU results from one of three events:

a) If the initiating SPM chooses not to use a SHORT CONNECT SPDU, an S-CONNECT request results in the assignment of a transport connection. When the transport connection is established, a CONNECT SPDU is sent on the transport normal flow.

b) If the initiating SPM chose to use a SHORT CONNECT SPDU, and sent the SHORT CONNECT SPDU on the User-data of the T-CONNECT request, the receipt of a T-CONNECT confirm with no SPDU in the User-data results in a CONNECT SPDU. This is sent on the transport normal flow.

c) If the initiating SPM chose to use a SHORT CONNECT SPDU (on a new or established transport connection), an incoming SHORT REFUSE SPDU with the reason code value indicating "unknown connection summary" results in a CONNECT SPDU. This is sent on the transport normal flow.

In all cases, if the Data Overflow ... *continue with rest of original paragraph.*

19) Subclause 7.4.1

Replace b) 1) with:

b) A Connect/Accept Item parameter group containing:

1) A Protocol Options parameter which enables the responder to indicate its ability to receive extended concatenated SPDUs whether the null encoding option is selected for use on this session connection, and its ability to receive the following short-form SPDUs – SHORT DATA TRANSFER, SHORT FINISH, SHORT DISCONNECT and SHORT ABORT SPDU.

The responder shall not select the null encoding option unless:

- the initiator indicates on the CONNECT SPDU that it is able to use the null encoding option; and
- the functional units selected for use on the session connection [see d) below] are precisely:
  - i) kernel functional unit;
  - ii) full-duplex functional unit;
  - iii) no-orderly-release functional unit.

20) Subclause 7.11

Add the following text at the end of the first sentence of this subclause:

Normal data is transferred by the DATA TRANSFER SPDU unless the null encoding option is selected, in which case the NULL SPDU is used (see 7.49). If the short encoding option is selected, the SPM may transfer data using the SHORT DATA TRANSFER SPDU (see 7.46).

21) Subclauses 7.38 through 7.41

Add the following new subclauses, numbered 7.38 through 7.41.3, after 7.37.3:

7.38 SHORT CONNECT SPDU

The SHORT CONNECT SPDU is sent as a protocol option at the choice of the initiating SPM to establish a session connection if the Session-user Requirements parameter in the S-CONNECT request consists, of only the kernel, full-duplex and no orderly release functional units, and there are no calling and called session selectors.

The SHORT CONNECT SPDU is transmitted by the initiator of the transport connection in order to initiate a session connection when the initiating SPM has chosen to use this SPDU. The SPDU can be transmitted on the User-data of a T-CONNECT request primitive or on a previously assigned, established transport connection.

The initiating SPM can choose to use the SHORT CONNECT SPDU if either:

a) an Upper-Layer Context Identifier specification is available that, with appropriate parameter values for the Upper-Layer Context Identifier, summarizes the CONNECT SPDU that would be sent if the initiating SPM chose not to use the SHORT CONNECT SPDU; or

b) in the parameters of the S-CONNECT request:

   i) the Session Requirements parameter requests only the kernel, full-duplex and no-orderly-release function units;

   ii) the Called Session Address and Calling Session Address have NIL values of the Called and Calling Session Selectors;

   iii) the Session Connection Identifier is absent.

7.38.1 Content of SHORT CONNECT SPDU

The SHORT CONNECT SPDU contains:

a) A Nested Connection Identifier parameter assigned to this connection for a nested session connection only.

b) A Connection-Summary parameter which identifies an Upper-Layer Context Specification that summarizes the values that would be contained in the parameters of a CONNECT SPDU that could be issued as a result of the S-CONNECT request. The semantic content of the User-data of the S-CONNECT request, if any, shall be included in the Connection-Summary via the Session-user-summary parameter of the S-CONNECT request.

   If the Upper-Layer Context Specification defines its own parameters, these shall be included in the Connection-Summary parameter.

c) A Special User-data parameter to transfer two bits of transparent data.

d) A User-information field to transfer transparent data. The User-information field shall not be present if any of the other parameters are present.

7.38.2 Sending the SHORT CONNECT SPDU

If the initiating SPM has chosen to use the SHORT CONNECT SPDU, an S-CONNECT request results in the assignment of a new or already established transport connection.

If the assigned transport connection is not yet established, and the SHORT CONNECT SPDU, including any parameter or User-information encodings, is small enough to be conveyed as User-data on the T-CONNECT request, the initiating SPM shall optionally send the SHORT CONNECT SPDU on the User-data parameter of the T-CONNECT request.

If the assigned transport connection is not yet established and either the SHORT CONNECT SPDU is too large for the User-data of the T-CONNECT request or the initiating SPM chooses, the initiating SPM shall wait until the transport connection is established, and then shall send the SHORT CONNECT SPDU on the transport normal flow.

If the assigned transport connection is already established, the SHORT CONNECT SPDU shall be sent on the transport normal flow.

7.38.3 Receiving the SHORT CONNECT SPDU

A valid incoming SHORT CONNECT SPDU which is acceptable to the receiving SPM results in an S-CONNECT indication to the SS-user. The SPM then waits for an S-CONNECT response from the called SS-user. If the receiving SPM is not able to handle the incoming request for a session connection, it does not issue any service primitive to the SS-user and formats a SHORT REFUSE SPDU (see 7.40.2).

If the receiving SPM does not support the short-connect option and the SHORT CONNECT SPDU is received as User-data on a T-CONNECT indication primitive, it shall ignore the SHORT CONNECT SPDU and shall not use the User-data parameter on the T-CONNECT response primitive.

A valid incoming SHORT CONNECT SPDU containing a Connection Summary parameter that the SPM is able to fully interpret, or containing no Connection Summary parameter, and that is acceptable to the receiving SPM results in an S-CONNECT indication to an SS-user, according to the Called Session Selector parameter that was summarized in the Connection Summary parameter (if present). The SPM then waits for an S-CONNECT response from the called SS-user.

A valid incoming SHORT CONNECT SPDU containing a Connection Summary parameter that identifies an Upper-Layer Context Identifier that is known to the SPM, but which cannot be fully interpreted results in the SPM sending a SHORT ACCEPT SPDU in which the completion field indicates the connection establishment is incomplete. The SPM then waits until it receives a SHORT CONNECT CONTINUE SPDU.

A valid incoming SHORT CONNECT SPDU, received on an established transport connection, and containing a Connection Summary parameter that identifies an Upper-Layer Context Identifier that is not known to the SPM, results in the SPM sending a SHORT REFUSE SPDU in which the reason code value is "unknown connection summary". The SPM then waits until it receives a CONNECT SPDU.

A valid incoming SHORT CONNECT SPDU, received on the User-data of T-CONNECT indication, and containing a Connection Summary parameter that identifies an Upper-Layer Context Identifier that is not known to the SPM, is ignored by the SPM. The T-CONNECT indication itself results in a T-CONNECT response with no SPDU in the User-data. The SPM then waits until it receives a CONNECT SPDU.

7.39 SHORT ACCEPT SPDU

An S-CONNECT (accept) response results in a SHORT ACCEPT SPDU if the responding SS-user selects in the Session user requirements parameter of the S-CONNECT response only the kernel, full-duplex and no orderly release functional units, and there is no responding session selector. A SHORT ACCEPT CONTINUE SPDU is first sent if the size restriction of the transport primitive does not allow to send the SHORT ACCEPT SPDU directly.

After this, the SPM enters the data transfer phase and can receive any session service request or SPDU allowed by the selected functional units.

An SPM receiving a SHORT CONNECT SPDU containing a Connection Summary parameter that the SPM is able to fully interpret, or containing no Connection Summary parameter, may accept a proposal to establish a session connection by transferring a SHORT ACCEPT SPDU (after receiving an S-CONNECT response primitive) to the initiator, on the same transport connection. A SHORT ACCEPT CONTINUE SPDU is first sent if the size restriction of the transport primitive does not allow to send the SHORT ACCEPT SPDU directly.

An SPM receiving a SHORT CONNECT SPDU that identifies an Upper-Layer Context Identifier that is known to the SPM, but which cannot be fully interpreted, results in the SPM sending a SHORT ACCEPT SPDU in which the completion field indicates the connection establishment is incomplete.

7.39.1 Content of SHORT ACCEPT SPDU

The SHORT ACCEPT SPDU contains:

a) A nested connection identifier parameter assigned to this connection for a nested session connection only.

b) A completion field that indicates whether the session connection establishment is complete; if it is not complete, a SHORT CONNECT CONTINUE is expected.

c) A Connection-Summary parameter which identifies an Upper-Layer Context Specification that summarizes the values that would be contained in the parameters of an ACCEPT SPDU that could be issued as a result of the S-CONNECT response. The content of the User-data of the S-CONNECT response, if any, shall be included in the Connection-Summary via the Session-user-summary parameter of the S-CONNECT response.

   If the Upper-Layer Context Specification defines its own parameters, these shall be included in the Connection-Summary parameter.

d) A Special User-data parameter to transfer two bits of transparent data.

e) A User-information field to transfer transparent data. The User-information field shall not be present if any of the other parameters are present.

7.39.2 Sending the SHORT ACCEPT SPDU

An S-CONNECT (accept) response results in a SHORT ACCEPT SPDU being sent.

Following an incoming SHORT CONNECT SPDU that resulted in an S-CONNECT indication, an S-CONNECT response results in a SHORT ACCEPT SPDU. If the SHORT CONNECT SPDU was received on the User-data of a T-CONNECT indication, and if the size of the SHORT ACCEPT SPDU meets the size constraint of the T-CONNECT response primitive, the SHORT ACCEPT SPDU is sent on the T-CONNECT response. If the SHORT CONNECT SPDU was received on the User-data of a T-CONNECT indication, and the size of the SHORT ACCEPT SPDU does not meet the size constraint of the T-CONNECT response primitive, a SHORT ACCEPT CONTINUE SPDU is sent on the T-CONNECT response and the SHORT ACCEPT SPDU is then sent on the transport normal flow. If the SHORT CONNECT SPDU was received on an established transport connection, the SHORT ACCEPT SPDU is sent on the transport normal flow of the same connection. In either case, after this successful session connection, the SPM enters the data transfer phase and can receive any service request or SPDU that is allowed by the selected functional units, protocol options and current token positions.

If any of the minor synchronize, major synchronize or resynchronize functional units are selected but the activity management functional unit is not selected, the SPM sets V(A) and V(M) to the Initial Serial Number proposed by the called SS-user, which is the serial number to be used for the first synchronization point. V(R) is set to zero. Vsc is set false.

If the symmetric synchronize functional unit is selected but the activity management functional unit is not selected, the SPM sets V(Ar) and V(Mr) to the Initial Serial Number proposed by the called SS-user, which is the serial number to be used for the first received synchronization point. The SPM sets V(As) and V(Ms) to the Second Initial Serial Number proposed by the called SS-user, which is the serial number to be used for the first synchronization point to be sent. V(Rs) and V(Rr) are set to zero.

If the activity management functional unit has been selected, Vact is set false.

An incoming SHORT CONNECT SPDU that identifies an Upper-Layer Context Identifier that is known to the SPM, but which cannot be fully interpreted, results in a SHORT ACCEPT SPDU in which the completion field indicates the connection establishment is incomplete. If the SHORT CONNECT SPDU was received on the User-data of a T-CONNECT indication, the SHORT ACCEPT SPDU is sent on the T-CONNECT response. If the SHORT CONNECT SPDU was received on an established transport connection, the SHORT ACCEPT SPDU is sent on the transport normal flow of the same connection. In either case, the SPM waits for a SHORT CONNECT CONTINUE SPDU.

7.39.3 Receiving the SHORT ACCEPT SPDU

A valid incoming SHORT ACCEPT SPDU in which the completion field indicates the connection establishment is complete, results in an S-CONNECT (accept) confirm. After this successful session connection, the SPM enters the data transfer phase and can receive any service request or SPDU that is allowed by the selected functional units, protocol options and current token positions.

If any of the minor synchronize, major synchronize or resynchronize functional units are selected but the activity management functional unit is not selected, the SPM sets V(A) and V(M) to the Initial Serial Number contained in the SHORT ACCEPT SPDU, which is the serial number to be used for the first synchronization point. V(R) is set to zero. Vsc is set false.

If the symmetric synchronize functional unit is selected but the activity management functional unit is not selected, the SPM sets V(As) and V(Ms) to the Initial Serial Number contained in the SHORT ACCEPT SPDU, which is the serial number to be used for the first synchronization point to be sent. The SPM sets V(Ar) and V(Mr) to the Second Initial Serial Number contained in the SHORT ACCEPT SPDU, which is the serial number to be used for the first received synchronization point. V(Rs) and V(Rr) are set to zero.

If the activity management functional unit has been selected, Vact is set false.

A valid incoming SHORT ACCEPT SPDU in which the completion field indicates the connection establishment is not complete, results in the SPM sending a SHORT CONNECT CONTINUE SPDU. The SPM then waits until it receives a SHORT ACCEPT CONTINUE SPDU or a SHORT REFUSE SPDU.

7.40 SHORT CONNECT CONTINUE SPDU

The SHORT CONNECT CONTINUE SPDU is used by the SPM to send the Summary parameters value for an Upper-Layer Context Identifier sent on a previous SHORT CONNECT SPDU.

7.40.1 Content of SHORT CONNECT CONTINUE SPDU

The SHORT CONNECT CONTINUE SPDU contains:

- Summary parameters containing the uncompressed parameter value as specified in the Upper-Layer Context specification identified by the Connection Summary parameter of the previous SHORT CONNECT SPDU.

7.40.2 Sending the SHORT CONNECT CONTINUE SPDU

A valid incoming SHORT ACCEPT SPDU in which the completion field indicates the connection establishment is not complete, results in the SPM sending a SHORT CONNECT CONTINUE SPDU. The SPM then waits until it receives a SHORT ACCEPT CONTINUE SPDU or a SHORT REFUSE SPDU.

7.40.3 Receiving the SHORT CONNECT CONTINUE SPDU

A valid incoming SHORT CONNECT CONTINUE SPDU which, with the previous SHORT CONNECT SPDU, is acceptable, results in an S-CONNECT indication to an SS-user, according to the Called Session Selector parameter that was summarized in the Connection Summary parameter of the SHORT CONNECT SPDU or the Summary parameter of the SHORT CONNECT CONTINUE SPDU. The SPM then waits for an S-CONNECT response from the called SS-user.

7.41 SHORT ACCEPT CONTINUE SPDU

The SHORT ACCEPT CONTINUE SPDU is used by the SPM to close the connection establishment phase when uncompressed parameters have been received (short-encoding option only) or to inform the sending SPM that the SHORT ACCEPT SPDU will be sent on the transport normal flow.

7.41.1 Content of SHORT ACCEPT CONTINUE SPDU

The SHORT ACCEPT CONTINUE SPDU contains:

a) A nested connection identifier parameter assigned to this connection for a nested session connection only.

b) A Connection-Summary parameter which identifies an Upper-Layer Context Specification that summarizes the values that would be contained in the parameters of an ACCEPT SPDU that could be issued as a result of the S-CONNECT response. The content of the User-data of the S-CONNECT response, if any, shall be included in the Connection-Summary via the Session-user-summary parameter of the S-CONNECT response.

   If the Upper-Layer Context Specification defines its own parameters, these shall be included in the Connection-Summary parameter.

c) A Special User-data parameter to transfer two bits of transparent data.

d) A User-information field to transfer transparent data. The User-information field shall not be present if any of the other parameters are present.

7.41.2 Sending the SHORT ACCEPT CONTINUE SPDU

An S-CONNECT response, following an incoming SHORT CONNECT CONTINUE SPDU that resulted in an S-CONNECT indication, results in a SHORT ACCEPT CONTINUE SPDU. This SPDU is sent to the transport normal flow. After this successful session connection, the SPM enters the data transfer phase and can receive any service request or SPDU that is allowed by the selected functional units, protocol options and current token positions.

An S-CONNECT response, following an incoming SHORT CONNECT SPDU in a T-CONNECT indication, results in a SHORT ACCEPT CONTINUE SPDU if the SHORT ACCEPT SPDU does not fit in the User-data parameter of the T-CONNECT response. The SHORT ACCEPT SPDU is then sent to the transport normal flow.

If any of the minor synchronize, major synchronize or resynchronize functional units are selected but the activity management functional unit is not selected, the SPM sets V(A) and V(M) to the Initial Serial Number proposed by the called SS-user, which is the serial number to be used for the first synchronization point. V(R) is set to zero. Vsc is set false.

If the symmetric synchronize functional unit is selected but the activity management functional unit is not selected, the SPM sets V(Ar) and V(Mr) to the Initial Serial Number proposed by the called SS-user, which is the serial number to be used for the first received synchronization point. The SPM sets V(As) and V(Ms) to the Second Initial Serial Number proposed by the called SS-user, which is the serial number to be used for the first synchronization point to be sent. V(Rs) and V(Rr) are set to zero.

If the activity management functional unit has been selected, Vact is set false.

7.41.3 Receiving the SHORT ACCEPT CONTINUE SPDU

A valid incoming SHORT ACCEPT CONTINUE SPDU following the sending of a SHORT CONNECT CONTINUE SPDU results in an S-CONNECT confirm. After this successful session connection, the SPM enters the data transfer phase and can receive any service request or SPDU that is allowed by the selected functional units, protocol options and current token positions.

On receipt of a valid incoming SHORT ACCEPT CONTINUE SPDU following the sending of a SHORT CONNECT SPDU, the SPM shall wait until it receives a SHORT ACCEPT SPDU on the transport normal flow.

After this successful connection, the SPM enters the data transfer phase and can receive any service request or SPDU that is allowed by the available functional units and current token positions. If any of the minor synchronize, major synchronize or resynchronize functional units are selected but the activity management functional unit is not selected, the SPM sets V(A) and V(M) to the Initial Serial Number contained in the SHORT ACCEPT SPDU, which is the serial number to be used for the first synchronization point. V(R) is set to zero. Vsc is set false.

If the symmetric synchronize functional unit is selected but the activity management functional unit is not selected, the SPM sets V(As) and V(Ms) to the Initial Serial Number contained in the SHORT ACCEPT SPDU, which is the serial number to be used for the first synchronization point to be sent. The SPM sets V(Ar) and V(Mr) to the Second Initial Serial Number contained in the SHORT ACCEPT SPDU, which is the serial number to be used for the first received synchronization point. V(Rs) and V(Rr) are set to zero.

If the activity management functional unit has been selected, Vact is set false.

7.42 SHORT REFUSE SPDU

A SHORT REFUSE SPDU is used by the responder (SS-user or SPM) to reject an attempt to establish a session connection with the SHORT CONNECT SPDU. A SHORT REFUSE CONTINUE SPDU is first sent if the size restriction of the transport primitive does not allow to send the SHORT REFUSE SPDU directly.

7.42.1 Content of SHORT REFUSE SPDU

The SHORT REFUSE SPDU contains, in order, a field that identifies if the transport connection may be retained, followed by zero, one or more parameter fields and, optionally, a User-information field.

The SHORT REFUSE SPDU contains:

a) A nested connection identifier parameter assigned to this connection for a nested session connection only.

b) A Transport Disconnect parameter which indicates whether or not the transport connection is to be kept.

c) A Reason Code parameter giving the reason for refusal of the attempt to establish a session connection.

d) A Summary-response parameter containing a value defined by the Upper-Layer Context specification identified in the previous SHORT CONNECT SPDU.

e) A User-information field which allows transparent User-data to be transferred.

7.42.2 Sending the SHORT REFUSE SPDU

An S-CONNECT (reject) response results in a SHORT REFUSE SPDU being sent as User-data on the transport connection T-CONNECT response primitive if the size of the SHORT REFUSE SPDU meets the size constraint of the T-CONNECT response primitive,. Otherwise, a SHORT REFUSE CONTINUE SPDU is sent on the T-CONNECT response and the SHORT REFUSE SPDU is then sent on the transport normal flow. The SHORT REFUSE SPDU may also be sent if the responding SPM is unable to handle the request for a session connection. No session connection is established.

An S-CONNECT (reject) response, following an S-CONNECT indication that resulted from incoming SHORT CONNECT SPDU or SHORT CONNECT CONTINUE SPDU, results in a SHORT REFUSE SPDU. If the SHORT CONNECT was received on the User-data of the T-CONNECT indication and no SHORT CONNECT CONTINUE has been received and if the size of the SHORT REFUSE SPDU meets the size constraint of the T-CONNECT response primitive, the SHORT REFUSE SPDU is sent on the User-data of the T-CONNECT response. If the size does not meet the size constraint of the T-CONNECT response primitive, a SHORT REFUSE CONTINUE SPDU is sent on the T-CONNECT response and the SHORT REFUSE is then sent on the transport normal flow.

If the SHORT CONNECT was received on an established transport connection or a SHORT CONNECT CONTINUE has been received, the SHORT REFUSE SPDU is sent on the transport normal flow.

If the Transport Disconnect parameter indicates that the transport connection can be reused, the SPM waits for a CONNECT SPDU or a SHORT CONNECT SPDU. Otherwise, the SPM starts the timer, TIM, and waits for a T-DISCONNECT indication. If the timer expires before receipt of a T-DISCONNECT indication, the SPM requests transport disconnection with a T-DISCONNECT request. The timer is canceled on receipt of a T-DISCONNECT indication.

NOTE – The value of TIM is a local implementation dependent matter, related to quality of service.

7.42.3 Receiving the SHORT REFUSE SPDU

A valid incoming SHORT REFUSE SPDU in which the reason code does not indicate "unknown connection summary" results in an S-CONNECT (reject) confirm, with the Responding Session Address parameter set to the value of the Called Session Address provided in the S-CONNECT request. No session connection is established. If the Transport Disconnect parameter indicates that the retention of the transport connection has been requested by the called SPM, and this is acceptable to the calling SPM, the SPM waits for an S-CONNECT request. Otherwise, the SPM releases the transport connection, by making a T-DISCONNECT request.

A valid incoming SHORT REFUSE SPDU in which the reason code indicates "unknown connection summary" results in the SPM sending a CONNECT SPDU.

7.43 SHORT REFUSE CONTINUE SPDU

The SHORT REFUSE CONTINUE SPDU is used by the SPM to inform the sending SPM that the SHORT REFUSE SPDU will be sent on the transport normal flow.

7.43.1 Content of SHORT REFUSE CONTINUE SPDU

The SHORT REFUSE CONTINUE SPDU does not contain any parameter.

7.43.2 Sending the SHORT REFUSE CONTINUE SPDU

An S-CONNECT (reject) response, following an incoming SHORT CONNECT SPDU in a T-CONNECT indication, results in a SHORT REFUSE CONTINUE SPDU if the SHORT REFUSE SPDU does not fit in the User-data parameter of the T-CONNECT response. The SHORT REFUSE SPDU is then sent to the transport normal flow.

7.43.3 Receiving the SHORT ACCEPT CONTINUE SPDU

On receipt of a valid incoming SHORT REFUSE CONTINUE SPDU following the sending of a SHORT CONNECT SPDU, the SPM shall wait until it receives a SHORT REFUSE SPDU on the transport normal flow.

7.44 SHORT FINISH SPDU

If the short-encoding protocol option has been selected, orderly release can be initiated by the transfer of a SHORT FINISH SPDU, which may be transferred during the data transfer phase. It requests as a response one of:

a) a SHORT DISCONNECT SPDU to complete the release of the session connection;

b) a DISCONNECT SPDU to complete the release of the session connection;

c) a NOT FINISHED SPDU to refuse the release of the session connection if the release token is available.

The SHORT FINISH SPDU is transferred in sequence with any normal data being transferred. The right to issue a SHORT FINISH SPDU is restricted to the owner of all available tokens.

7.44.1 Content of SHORT FINISH SPDU

The SHORT FINISH SPDU contains:

a) a nested connection identifier parameter assigned to this connection for a nested session connection only;

b) a Transport Disconnect parameter which indicates whether or not the transport connection is to be kept, subject to the restrictions specified in 6.2.4;

c) a User-information field which allows transparent User-data to be transferred.

7.44.2 Sending SHORT FINISH SPDU

If the short-encoding protocol option is selected, an S-RELEASE request results, at the SPM's option, in a SHORT FINISH SPDU or a FINISH SPDU. The SHORT FINISH SPDU is sent on the transport normal flow.

After transferring a SHORT FINISH SPDU, the SPM may not send any further SPDUs (except SHORT ABORT SPDU or, in the case of collision of SHORT FINISH SPDUs, a SHORT DISCONNECT SPDU) unless a NOT FINISHED SPDU or a RESYNCHRONIZE SPDU is received, after which the data transfer phase may be resumed. Receipt of a SHORT DISCONNECT SPDU signals completion of orderly session release.

7.44.3 Receiving SHORT FINISH SPDU

A valid incoming SHORT FINISH SPDU results in an S-RELEASE indication. The User-information is passed to the SS-user. The SPM waits for an S-RELEASE response.

7.45 SHORT DISCONNECT SPDU

If the short-encoding protocol option has been selected, after receipt of a SHORT FINISH SPDU or a FINISH SPDU, a SHORT DISCONNECT may be transferred. Receipt of a SHORT DISCONNECT SPDU after transferring a SHORT FINISH SPDU or a FINISH SPDU signals the orderly release of the session connection. The SHORT DISCONNECT SPDU is transferred in sequence with any normal data being transferred.

7.45.1 Content of SHORT DISCONNECT SPDU

The SHORT DISCONNECT SPDU contains:

a) a Nested Connection Identifier parameter assigned to this connection for a nested session connection only;

b) a User-information field which allows transparent User-data to be transferred.

7.45.2 Sending SHORT DISCONNECT SPDU

If the short-encoding protocol option is selected, an S-RELEASE response results, at the SPM's option, in a SHORT DISCONNECT SPDU or a DISCONNECT SPDU. The SHORT DISCONNECT SPDU is sent on the transport normal flow.

If the SHORT FINISH SPDU or FINISH SPDU indicated that the transport connection is to be kept for reuse and this is acceptable, the SPM waits for a CONNECT SPDU or a SHORT CONNECT SPDU. Otherwise, the SPM starts the timer, TIM, and waits for a T-DISCONNECT indication. If the timer expires before receipt of a T-DISCONNECT indication, the SPM requests transport disconnection with a T-DISCONNECT request. The timer is canceled on receipt of a T-DISCONNECT indication.

7.45.3 Receiving SHORT DISCONNECT SPDU

A valid incoming SHORT DISCONNECT SPDU results in an S-RELEASE (accept) confirm. The session connection ceases to exist.

If the transport connection is to be kept for reuse (see 6.2.4) the SPM waits for a suitable S-CONNECT request. Otherwise, a T-DISCONNECT request is issued.

7.46 SHORT DATA TRANSFER SPDU

If the short-encoding protocol option is selected, normal data may be transferred by use of the SHORT DATA TRANSFER SPDU.

The right to issue a SHORT DATA TRANSFER SPDU is subject to the token restrictions specified in Table 5.

7.46.1 Content of SHORT DATA TRANSFER SPDU

The SHORT DATA TRANSFER SPDU contains:

a) a User-information field to transfer transparent User-data whose maximum size is unlimited when segmenting has not been selected and whose maximum size is limited by the maximum TSDU size when segmenting has been selected;

b) a Special User-data parameter to transfer two bits of data.

7.46.2 Sending the SHORT DATA TRANSFER SPDU

If the short-encoding protocol option is selected, and at the SPM's option, an S-DATA request results in a SHORT DATA TRANSFER SPDU, provided, if segmenting has been selected, the SSDU is small enough for the SPDU to fit in a TSDU.

7.46.3 Receiving the SHORT DATA TRANSFER SPDU

A valid incoming SHORT DATA TRANSFER SPDU results in an S-DATA indication.

7.47 SHORT ABORT SPDU

The SHORT ABORT SPDU is used to cause, at any time, abnormal release of session connection on which the short-encoding protocol option is selected. This SPDU may also be used to release such a session connection when a protocol error is detected. The SHORT ABORT SPDU may or may not request that the transport connection be released by the receiving SPM. Use of the SHORT ABORT SPDU may result in loss of data.

7.47.1 Content of SHORT ABORT SPDU

The SHORT ABORT SPDU contains:

a) a nested connection identifier parameter assigned to this connection for a nested session connection only;

b) a Transport Disconnect parameter which indicates whether or not the transport connection is to be kept;

c) a Reason Code which gives the reason for the abort;

d) a User-information field which allows transparent data to be transferred.

7.47.2 Sending the SHORT ABORT SPDU

If the short-encoding protocol option has been selected, an S-U-ABORT request in which the SS-user data does not exceed 9 octets, or the detection of a protocol error in any state of the SPM, results, at the SPM's option, in a SHORT ABORT SPDU or one or more ABORT SPDUs, as specified in 7.9.2.

The SHORT ABORT SPDU is sent on the transport expedited flow, if it is available to the session connection. If the transport expedited flow is not available, the SHORT ABORT SPDU shall be sent on the transport normal flow.

The SPM starts the timer, TIM, and waits for an ABORT ACCEPT SPDU or a T-DISCONNECT indication. Any other SPDUs are discarded. If the timer expires before receipt of an ABORT ACCEPT SPDU or a T-DISCONNECT indication, the SPM requests transport disconnection with a T-DISCONNECT request. The timer is canceled on receipt of a T-DISCONNECT indication.

7.47.3 Receiving the SHORT ABORT SPDU

A valid incoming SHORT ABORT SPDU results in an S-U-ABORT indication or S-P-ABORT indication, depending on whether the abort is user-generated or provider-generated. The session connection ceases to exist. If the Transport Disconnect parameter in the received SHORT ABORT SPDU indicates that the transport connection is to be kept, and this is acceptable to the receiving SPM, an ABORT ACCEPT SPDU is sent. If the Transport Disconnect parameter in the received SHORT ABORT SPDU indicates the transport connection is not to be kept for reuse, or reuse of the transport connection is not acceptable to the receiving SPM, the receiving SPM releases the transport connection by issuing a T-DISCONNECT request.

7.48 Connection release when the null-encoding option is selected

When the null encoding option is selected, the session connection is only terminated by the termination of the supporting transport connection. No ABORT SPDU is transmitted, but the User-data of the S-U-ABORT request is sent in the User-data of the T-DISCONNECT service.

NOTE – The Transport provider imposes restrictions on the maximum size of the T-DISCONNECT User-data. If the S-U-ABORT request SS-user-data parameter will not fit, it is not transmitted.

7.48.1 Initiating termination

An S-U-ABORT request or the detection of a protocol error results in a T-DISCONNECT request.

In the case of an S-U-ABORT request, the User-data parameter of the T-DISCONNECT User-data shall have a first octet containing the value 1. If the SS-user-data parameter is present on the S-U-ABORT request, this shall be placed in the second and subsequent octets of the T-DISCONNECT request User-data, provided the entire value will fit within the restrictions imposed by the Transport provider. If the value will not fit in the T-DISCONNECT User-data, the T-DISCONNECT User-data parameter shall contain only the single octet of value 1 or shall be empty.

In the case of detection of a protocol error, or other problem internal to the SPM, the User-data parameter of the T-DISCONNECT shall be empty, or the first octet shall not have the value 1.

7.48.2 Receiving T-DISCONNECT

An incoming T-DISCONNECT indication with a User-data field in which the first octet has the value 1 results in an S-U-ABORT indication. If the T-DISCONNECT User-data field is longer than one octet, the second and subsequent octets shall be the SS-user-data parameter of the S-U-ABORT indication.

An incoming T-DISCONNECT indication with an empty User-data field or a User-data field in which the first octet is any value other than 1, results in an S-P-ABORT indication with no parameters.

7.49 NULL SPDU

When the null encoding option is selected, normal data is transferred by use of the NULL SPDU.

7.49.1 Content of NULL SPDU

The NULL SPDU contains a User-information Field to transfer transparent User-data whose maximum size is unlimited.

NOTE – Segmentation does not occur with the null-encoding option. Since null-encoding is only used when the kernel, no-orderly-release and full-duplex functional units are the only functional units selected, there is no data token.

7.49.2 Sending the NULL SPDU

An S-DATA request, when the null encoding option has been selected, results in a NULL SPDU.

7.49.3 Receiving the NULL SPDU

When the null encoding option is selected, a received T-DATA indication shall be treated as the receipt of an incoming NULL SPDU, and results in an S-DATA indication.

22) Subclause 8.2

Change title of 8.2 as shown below:

8.2 SPDU structure (long form SPDUs)

Change, based on the additions, the first sentence of 8.2 as follows:

This subclause specifies the general structure of long-form SPDUs in terms of their constituent fields. Long-form SPDUs are all SPDUs whose names do not begin with "SHORT".

Add "long-form" *in front of* "SPDU" *throughout the rest of 8.2.*

Add the following paragraph after the first paragraph of 8.2:

The NULL SPDU does not use this structure (see 8.2.9).

23) Subclauses 8.2.1 through 8.2.8

No change, except for the addition of the phrase "long-form" *in front the word* "SPDU" *wherever it occurs in these subclauses.*

Add the following new subclause after 8.2.8:

8.2.9 Structure of NULL SPDU

The NULL SPDU consists only of a User-information field. There is no SI or LI field.

NOTE – The NULL SPDU maps an outgoing SSDU unchanged as a TSDU, and maps an incoming TSDU unchanged as an SSDU.

24) Subclauses 8.5 and 8.6

Add two new subclauses after 8.4 to be called 8.5 and 8.6:

8.5 SPDU structure (short-form SPDUs)

This subclause specifies the general structure of short-form SPDUs in terms of their constituent fields. Short-form SPDUs are all SPDUs whose names begin with "SHORT". The general structure for such SPDUs is illustrated in Table 47.

Table 47 – Short-form SPDU structure

Short-form SPDUs	SI&P field	Parameter field	User-information field
SI&P octet	iiiiipxx		

Short-form SPDUs shall contain, in the following order:

a) An SI&P field of one octet.

b) Zero, one or more parameter fields defined for the short-form SPDU.

c) Either:

   i) one unspecified-length parameter, if defined for the SPDU; or

   ii) the User-information field, if defined for the SPDU and if present.

The SI&P octet contains the following fields:

d) The SI field in bits 4-8, shown as "i" in Table 47. This identifies the type of short-form SPDU; the high-order bit (bit 8 of the SI&P octet) is 1 for all short-form SPDUs.

e) The parameter indication in bit 3, shown as "p" in Table 47 indicates whether the first octets following the SI&P octet are SPDU parameters.

f) The parameters or special data field in bits 1 and 2, shown as "x" in Table 47. These bits have different meanings for different short-form SPDUs.

The overall length of a short-form SPDU is determined by the TSDU length.

8.5.2 Parameter indication

Bit 3 of the SI&P octet indicates whether the octet, if any, immediately following the SI&P octet contains SPDU parameters. If bit 3 indicates there are no parameters in the following octet, and the short-form SPDU has a length greater than one octet, all octets after the first are User-information.

The encoding of this bit shall be:

a) bit 3 = 0: octet 2 and any following octets are User-information;

b) bit 3 = 1: at least octet 2 contains an SPDU parameter.

If there are no following octets, bit 3 shall be zero.

It is specified for each short-form SPDU, if octet 2 contains an SPDU parameter, how many of the following octets contain parameters and which of the following octets, if any, contain User-information.

8.5.3 Special data

For some short-form SPDUs bits 1 and 2 of the SI&P octet encode a Special User-data value. Subclauses of 8.6 specify which short-form SPDUs use these bits for a Special User-data value.

If the value of the Special User-data parameter has a length of two bits, the high-order bit shall be placed in bit 2 of the SI&P octet and the low-order bit shall be placed in bit 1.

If the value of the Special User-data parameter has a length of one bit, this shall be placed in bit 1 of the SI&P octet and bit 2 of the SI&P octet shall be zero.

If there is no Special User-data parameter on the invoking service, both bit 1 and bit 2 shall be zero.

8.5.4 Parameters in SI&P octet

For some short-form SPDUs, bits 1 and 2 of the SI&P octet encode one or two parameters. The use of these bits is specified in the appropriate subclause of 8.6.

8.5.5 Parameters in following octets

The encoding of parameters in the octets following the SI&P octet is specific to each short-form SPDU.

Parameters of short-form SPDUs are either of fixed length defined in 8.6 for that SPDU, or have an unspecified length.

A parameter with an unspecified length is always the last parameter in the encoding of the SPDU. The end of the parameter is at the end of the SPDU. If a particular short-form SPDU contains a parameter of unspecified length, it will not contain a User-information field.

8.5.6 User-information fields

Following the SI&P octet and any fixed length parameters, the User-information field of the short-form SPDU shall contain the SSDU. The order of the octets and the order of the bits in the SSDU shall be maintained in the SPDU.

8.6 Short-form SPDU identifiers and associated parameter fields

All short-form SPDUs have SI fields of 5 bits, in which the high-order bit (bit 8 of the SI&P field) is 1.

The short-form SPDUs specified in the remainder of this subclause have SI fields of five bits. These are represented in this subclause as bit strings, using the notation:

"VWXYZ"b

where each of "V", "W", "X", "Y", "Z" is either a "0" or "1". In the SI&P octet of the short-form SPDU:

- the value shown in position V represents bit 8 of the SI&P octet;
- the value shown in position W represents bit 7 of the SI&P octet;
- the value shown in position X represents bit 6 of the SI&P octet;
- the value shown in position Y represents bit 5 of the SI&P octet;
- the value shown in position Z represents bit 4 of the SI&P octet.

NOTE – Bit 8 (V) is 1 for all short-form SPDUs (and bit 8 is always zero in the SI octet of a long-form SPDU). For most, but not all, short-form SPDUs, WXYZ is the same as the low-order four bits of the SI of the corresponding long-form SPDU.

8.6.1 SHORT CONNECT (SCN) SPDU

8.6.1.1 The SI field of the SI&P octet shall contain "11101"b.

8.6.1.2 Bit 3 of the SI&P octet shall contain the parameter indication, as specified in 8.5.2.

8.6.1.3 Bits 1 and 2 of the SI&P octet shall contain the Special User-data field. The encoding is specified in 8.5.3.

8.6.1.4 If the parameter indication in bit 3 of the SI&P is one, octet 2 and any following octets shall contain parameters as specified in Table 48.

Table 48 – SCN SPDU Structure

Parameter	m/nm	Presence indication	Length	Reference
Presence and length (P&L)	m	SI&P:3	1	8.6.1.5
Nesting identifier	nm	P&L:8	1	8.6.1.6
Reserved_one	nm	P&L:7	1	8.6.1.7
Reserved_two	nm	P&L:6	1	8.6.1.7
Connection-Summary	nm	SI&P:3	Unspecified	8.6.1.9

8.6.1.5 The Presence and length parameter indicates which of the four non-mandatory parameters are present and the type and length of the identifier portion of the Connection Summary parameter. Bits 6 to 8 shall indicate which of the non-mandatory parameters are present. The encoding shall be:

a) bit 8 = 1: the Nesting identifier parameter is present;

b) bit 7 = 1: the reserved-one parameter is present;

c) bit 6 = 1: the reserved-two parameter is present.

Bit 5 is reserved.

Bits 1-4 shall indicate the type and length of the Upper-Layer Context Identifier in the Connection Summary parameter. The encoding shall be:

a) All of bits 1-4 zero: The Upper-Layer Context Identifier is a Restricted-form identifier represented in the first two octets of the Connection Summary parameter.

b) At least one bit of 1-4 is non-zero: The Upper-Layer Context Identifier is a Global-form identifier. Bits 1-4 of the P&L parameter represent a binary number. The length of the Upper-Layer Context Identifier is this binary number plus 4, and is represented in the leading octets of the Connection Summary parameter.

NOTE – This allows the global form of the identifier (a BER-encoded ASN.1 Object Identifier) to have any length in the range 5 to 19.

8.6.1.6 The nested session identifier shall be as defined in ISO/IEC 8327-1/PDAM1. This parameter shall be absent if the nested session identifier is zero.

8.6.1.7 The Reserved_one and Reserved_two parameters are reserved for future use. They shall not be present when the SHORT CONNECT SPDU is sent, and shall be ignored on a SHORT CONNECT SPDU that is received.

8.6.1.9 The Connection Summary parameter shall contain the identifier and parameters of an Upper-Layer Context Specification. The identifier shall be either a Restricted-form identifier or a Global-form identifier.

A Restricted-form identifier shall be an integer in the range 1 to 65535. It shall be encoded as binary number in the first two octets of the Connection Summary parameter.

NOTE – The Restricted-form is not globally unambiguous. The assignment of an integer identifier to an Upper-Layer Context specification is by bilateral agreement, although this "bilateral" agreement may be administered centrally for some community of Open systems.

A Global-form identifier shall be the encoding, according to the Basic Encoding Rules for ASN.1, of an Object Identifier that identifies the Upper-Layer Context specification.

The parameters of the Upper-Layer Context Specification, if any, are represented in the octets after the identifier.

If the Upper-Layer Context Specification has both compressed and uncompressed forms of its parameters, these may be present in either form, at sender's option.

8.6.1.10 If the parameter indication in bit 3 of the SI&P is zero, octet 2, if present, and any following octets shall contain the User-information field.

8.6.2 SHORT ACCEPT (SAC) SPDU

8.6.2.1 The SI field shall contain "11110"b.

8.6.2.2 Bit 3 of the SI&P octet shall contain the parameter indication, as specified in 8.5.2.

8.6.2.3 Bits 1 and 2 of the SI&P octet shall contain the Special User-data field. The encoding is specified in 8.5.3.

8.6.2.4 If the parameter indication in bit 3 of the SI&P is one, octet 2 and all following octets shall contain parameters as specified in Table 49.

Table 49 – SAC SPDU Structure

Parameter	m/nm	Presence indication	Length	Reference
Presence and length (P&L)	m	SI&P:3	1	8.6.2.5
Nesting identifier	nm	P&L:8	1	8.6.2.6
Connection-Summary	nm	SI&P:3 and SPDU continues after Nesting-identifier (or P&L)	Unspecified	8.6.2.9

8.6.2.5 The Presence and length parameter indicates whether the Nesting identifier parameter is present, whether a SHORT CONNECT CONTINUE SPDU is expected and the type and length of the identifier portion of the Connection Summary parameter, if present.

Bit 8 shall indicate whether the Nesting identifier parameter is present. The encoding shall be:

a) bit 8 = 1: the Nesting identifier parameter is present;

b) bit 8 = 0: the Nesting identifier parameter is not present.

Bit 7 shall indicate whether responder expects a SHORT CONNECT CONTINUE SPDU or has completed the establishment of the connection. The encoding shall be:

a) bit 7 = 1: Connection establishment is not complete; SHORT CONNECT CONTINUE SPDU expected;

b) bit 7 = 0: Connection establishment completed; no SHORT CONNECT CONTINUE SPDU expected.

Bits 5 and 6 are reserved.

Bits 1-4 shall indicate the type and length of the Upper-Layer Context Identifier in the Connection Summary parameter, if that parameter is present. The encoding shall be:

a) All of bits 1-4 zero: The Upper-Layer Context Identifier is a Restricted-form identifier represented in the first two octets of the Connection Summary parameter.

b) At least one bit of 1-4 is non-zero: The Upper-Layer Context Identifier is a Global-form identifier. Bits 1-4 of the P&L parameter represent a binary number. The length of the Upper-Layer Context Identifier is this binary number plus 4, and is represented in the leading octets of the Connection Summary parameter.

NOTE – This allows the global form of the identifier (a BER-encoded ASN.1 Object Identifier) to have any length in the range 5 to 19.

If no Connection Summary parameter is present, bits 1 to 4 of the P&L parameter shall be zero.

8.6.2.6 The nested session identifier shall be as defined in ISO/IEC 8327-1/PDAM1. This parameter shall be absent if the nested session identifier is zero.

8.6.2.7 The presence of the Connection Summary parameter is determined by the length of the SPDU. Any octets after the Nesting identifier (if present), or after the P&L parameter (if bit 8 of that parameter is zero) are the Connection Summary parameter.

If the Connection Summary parameter is present, it shall contain the identifier and parameters of an Upper-Layer Context Specification. The identifier shall be either a Restricted-form identifier or a Global-form identifier.

A Restricted-form identifier shall be an integer in the range 1 to 65535. It shall be encoded as binary number in the first two octets of the Connection Summary parameter.

NOTE – The Restricted-form used on the SHORT ACCEPT SPDU identifies the response within the scope of the Upper-Layer Context specification referenced by the previous SHORT CONNECT SPDU. It may therefore be globally unambiguous, if the SHORT CONNECT SPDU used a Global-form identifier.

A Global-form identifier shall be the encoding, according to the Basic Encoding Rules for ASN.1, of an Object Identifier that identifies the Upper-Layer Context specification.

The parameters of the Upper-Layer Context Specification, if any, are represented in the octets after the identifier.

If the Upper-Layer Context Specification has both compressed and uncompressed forms of its parameters, these may be present in either form, at sender's option.

8.6.2.8 If the parameter indication in bit 3 of the SI&P is zero, octet 2, if present, and any following octets shall contain the User-information field.

8.6.3 SHORT CONNECT CONTINUE (SCNC) SPDU

8.6.3.1 The SI field of the SI&P octet shall contain "11111"b.

8.6.3.2 Bit 3 of the SI&P octet shall contain the parameter indication, as specified in 8.5.2.

8.6.3.3 Bits 1 and 2 of the SI&P octet are reserved. They shall be zero when the SPDU is sent and shall be ignored when the SPDU is received.

8.6.3.4 If the parameter indication in bit 3 of the SI&P is one, octet 2 and all following octets shall contain parameters as specified in Table 50.

Table 50 – SCNC SPDU Structure

Parameter	Length	Octets	Reference
Summary-Parameters	Unspecified	2-end	8.6.3.5

8.6.3.5 The Summary-Parameters parameter, if present, shall contain the uncompressed parameter specified for the Upper-Layer Context specification identified by the Connection Summary parameter of the previous SHORT CONNECT SPDU.

8.6.3.6 If the parameter indication in bit 3 of the SI&P is zero, octet 2, if present, and any following octets shall contain the User-information field.

8.6.4 SHORT ACCEPT CONTINUE (SACC) SPDU

8.6.4.1 The SI field of the SI&P octet shall contain "11011"b.

8.6.4.2 Bit 3 of the SI&P octet shall contain the parameter indication, as specified in 8.5.2.

8.6.4.3 Bits 1 and 2 of the SI&P octet are reserved. They shall be zero when the SPDU is sent and shall be ignored when the SPDU is received.

8.6.4.4 If the parameter indication in bit 3 of the SI&P is one, octet 2 and all following octets shall contain parameters as specified in Table 51.

Table 51 – SACC SPDU Structure

Parameter	Length	Octets	Reference
Summary-Parameters	Unspecified	2-end	8.6.4.5

8.6.4.5 The Summary-Parameters parameter, if present, shall contain the uncompressed parameter specified for the Upper-Layer Context specification identified by the Connection Summary parameter of the previous SHORT ACCEPT SPDU.

8.6.4.6 If the parameter indication in bit 3 of the SI&P is zero, octet 2, if present, and any following octets shall contain the User-information field.

8.6.5 SHORT REFUSE (SRF) SPDU

8.6.5.1 The SI field shall contain "11100"b.

8.6.5.2 Bit 3 of the SI&P octet shall contain the parameter indication, as specified in 8.5.2.

8.6.5.3 Bit 2 of the SI&P octet shall contain Transport Disconnect field. This shall indicate whether or not the transport connection is to be kept. The encoding of this field shall be:

a) bit 2 = 1: transport connection is released;

b) bit 2 = 0: transport connection is retained.

8.6.5.4 Bit 1 of the SI&P octet shall indicate whether the refusal is transient or permanent. The encoding of this field shall be:

a) bit 1 = 1: rejection may be reported to calling SS-user as persistent;

b) bit 1 = 0: rejection may be reported to the calling SS-user as transient.

8.6.5.5 If the parameter indication in bit 3 of the SI&P is one, octet 2 and all following octets shall contain parameters as specified in Table 52.

Table 52 – SRF SPDU Structure

Parameter	m/nm	Presence indication	Length	Reference
Presence and length (P&L)	m	SI&P:3	1	8.6.1.5
Nesting identifier	nm	P&L:8	1	8.6.1.6
Summary-response	nm	SI&P:3 and SPDU continues after Nesting-identifier (or P&L)	Unspecified	8.6.1.9

8.6.5.6 The Presence and length parameter indicates whether the Nesting identifier parameter is present.

Bit 8 shall indicate whether the Nesting identifier parameter is present. The encoding shall be:

a) bit 8 = 1: the Nesting identifier parameter is present;

b) bit 8 = 0: the Nesting identifier parameter is not present.

Bit 7 shall indicate whether the responder supports (recognizes) the identifier of the Upper-Layer Context specification in the Connection-Summary on the SHORT CONNECT SPDU. The encoding shall be:

a) bit 7 = 1: Connection-Summary parameter on the SHORT CONNECT SPDU was recognized;

b) bit 7 = 0: Connection-Summary parameter on the SHORT CONNECT SPDU was not recognized.

Bits 1 to 6 are reserved.

8.6.5.7 The Summary-response parameter shall only be present if bit 7 of the P&L parameter is 1. If present, the Summary-response parameter shall contain a value specified by the Upper-Layer Context specification identified by the Connection Summary parameter of the previous SHORT CONNECT SPDU.

8.6.5.8 If the parameter indication in bit 3 of the SI&P is zero, octet 2, if present, and any following octets shall contain the User-information field.

8.6.6 SHORT FINISH (SFN) SPDU

8.6.6.1 The SI field of the SI&P octet shall contain "11001"b.

8.6.6.2 Bit 3 of the SI&P octet shall be zero.

8.6.6.3 Bit 2 of the SI&P octet shall contain the Transport Disconnect field. This shall indicate whether or not the transport connection is to be kept. The encoding for this field shall be:

a) bit 2 = 0: transport connection is released;

b) bit 2 = 1: transport connection is kept.

8.6.6.4 Bit 1 of the SI&P octet is reserved. It shall be zero when the SPDU is sent and shall be ignored when the SPDU is received.

8.6.6.5 Octet 2, if present, and any following octets shall contain the User-information field.

8.6.7 SHORT DISCONNECT (SDN) SPDU

8.6.7.1 The SI field of the SI&P octet shall contain "11010"b.

8.6.7.2 Bit 3 of the SI&P octet shall be zero.

8.6.7.3 Bit 2 of the SI&P octet shall contain the Transport Disconnect field. This shall indicate whether or not the transport connection is to be kept. The encoding for this field shall be:

a) bit 2 = 0: transport connection is released;

b) bit 2 = 1: transport connection is kept.

8.6.7.4 Bit 1 of the SI&P octet is reserved. It shall be zero when the SPDU is sent and shall be ignored when the SPDU is received.

8.6.7.5 Octet 2, if present, and any following octets shall contain the User-information field.

8.6.8 SHORT DATA TRANSFER (SDT) SPDU

8.6.8.1 The SI field shall contain "10001"b.

8.6.8.2 Bit 3 of the SI&P octet shall be zero.

8.6.8.3 Bits 1 and 2 of the SI&P octet shall contain the Special User-data field. The encoding is specified in 8.5.3.

8.6.8.4 Octet 2 and any following octets shall contain the User-information field.

NOTE – The SDT always has a length greater than one.

8.6.9 SHORT ABORT (SAB) SPDU

8.6.9.1 The SI field of the SI&P octet shall contain "10110"b.

8.6.9.2 Bit 3 of the SI&P octet shall contain the parameter indication, as specified in 8.5.2.

8.6.9.3 Bit 2 of the SI&P octet shall contain the Transport Disconnect field. This shall indicate whether or not the transport connection is to be kept. The encoding for this field shall be:

a) bit 2 = 0: transport connection is released;

b) bit 2 = 1: transport connection is kept.

8.6.9.4 Bit 1 of the SI&P octet is reserved. It shall be zero when the SPDU is sent and shall be ignored when the SPDU is received.

8.6.9.5 If the parameter indication in bit 3 of the SI&P is one, octet 2 shall contain the parameter specified in Table 53.

Table 53 – SAB SPDU Structure

Parameter	Length	Octets	Reference
Reason code	1	2	8.6.4.5

8.6.9.6 The Reason code parameter, if present, shall indicate the reason for the abort. The encoding of this field shall be:

a) bit 2 = 1: user abort;

b) bit 3 = 1: protocol error;

c) bit 4 = 1: no reason;

d) bit 5 = 1: implementation restriction stated in the PICS.

Bits 1, 6, 7 and 8 are reserved.

8.6.9.7 If the parameter indication in bit 3 of the SI&P is one and the Reason code field indicates user abort, then octet 3, if present, and any subsequent octets shall contain the User-information field.

8.6.9.8 If the parameter indication in bit 3 of the SI&P is one and the Reason code field does not indicate user abort, there shall be no User-information field.

8.6.9.9 If the parameter indication in bit 3 of the SI&P is zero, then octet 2, if present, and any subsequent octets shall contain the User-information field.

8.6.10 SHORT REFUSE CONTINUE (SRFC) SPDU

8.6.10.1 The SI field of the SI&P octet shall contain "10100"b.

8.6.10.2 Bit 3 of the SI&P octet shall contain the parameter indication, as specified in 8.5.2. As there are no parameters in the SPDU, bit 3 shall be set to zero.

8.6.10.3 Bits 1 and 2 of the SI&P octet are reserved. They shall be zero when the SPDU is sent and shall be ignored when the SPDU is received.

8.6.10.4 Octet 2, if present, and any following octets shall contain the User-information field.

25) Subclause 9.1.3

Add to the list after item b), as follows:

c) whether the null encoding option is supported;

d) whether the short-encoding option is supported.

Consequently, the existing item c) becomes item e).

26) Subclause A.5.1

Add to the right-hand side of the definition of fu-dom, *the additional term* NOR, *as follows:*

fu-dom = {FD, HD, EXCEP, TD, NR, SY, SS, DS, MA, RESYN, EX, ACT, CD, NOR}

Add to the list of functional units immediately following the following definition for NOR*:*

NOR No orderly release functional unit

27) Subclause A.5.4.20

Add the following rows to Table A.1 and change, as shown by the text, the descriptions of TCONind *and* TCONcnf*:*

Abbreviated name	Category	Name and description
NL	SPDU	NULL SPDU
SAB	SPDU	SHORT ABORT SPDU
SAC-TC-y	SPDU	SHORT ACCEPT (completed) SPDU in User-data of T-CONcnf
SAC-TC-n	SPDU	SHORT ACCEPT (not completed) SPDU in User-data of T-CONcnf
SAC-TD-y	SPDU	SHORT ACCEPT (completed) SPDU on T-DATAind
SAC-TD-n	SPDU	SHORT ACCEPT (not completed) SPDU on T-DATAind
SACC-TD	SPDU	SHORT ACCEPT CONTINUE SPDU on T-DATAind
SACC-TC	SPDU	SHORT ACCEPT CONTINUE SPDU in User-data of T-CONcnf
SRFC-TC	SPDU	SHORT REFUSE CONTINUE SPDU in User-data of T-CONcnf
SCNC	SPDU	SHORT CONNECT CONTINUE SPDU
SCN-TC	SPDU	SHORT CONNECT SPDU on User-data of T-CON ind (see Note 3)
SCN-TD	SPDU	SHORT CONNECT SPDU on T-DATAind
SDN	SPDU	SHORT DISCONNECT SPDU
SDT	SPDU	SHORT DATA TRANSFER SPDU
SFN	SPDU	SHORT FINISH SPDU
SRF-TC-nr	SPDU	SHORT REFUSE SPDU (with no reuse of transport connection) on User-data of T-CONcnf
SRF-TC-r	SPDU	SHORT REFUSE SPDU (with reuse of transport connection) on User-data of T-CONcnf
SRF-TD-nr	SPDU	SHORT REFUSE SPDU (with no reuse of transport connection) on T-DATAind
SRF-TD-r	SPDU	SHORT REFUSE SPDU (with reuse of transport connection) on T-DATAind

Also in Table A.1, change the descriptions of the following events:

TCONind	TS-provider	T-CONNECTind with no SPDU on the User-data
TCONcnf	TS-provider	T-CONNECTcnf with no SPDU on the User-data

Add Note 3:

3 If the short-connect protocol option is not supported, receipt of a T-CONNECT indication with a SHORT CONNECT SPDU in the User-data is treated as event TCONind.

Add the following rows to Table A.2:

Abbreviated Name	Name and description
STA01E	Await SAC in User-data of TCONrsp
STA01F	Await SCONrsp (after receiving SCN in User-data of TCONind)
STA02C	Await SAC on T-DATAind
STA02D	Await SACC
STA07	Await SCNC
STA08B	Await SCONrsp (after receiving SCN in User-data of T-DATAind)
STA700	Data transfer (null-encoding)

Add the following new rows to Table A.3, and change the descriptions of TCONreq *and* TCONrsp*:*

Abbreviated name	Category	Name and description
NL	SPDU	NULL SPDU
SAB	SPDU	SHORT ABORT SPDU
SAC-TC-y	SPDU	SHORT ACCEPT (completed) SPDU in User-data of T-CONrsp
SAC-TC-n	SPDU	SHORT ACCEPT (not completed) SPDU in User-data of T-CONrsp
SAC-TD-y	SPDU	SHORT ACCEPT (completed) SPDU on T-DATAreq
SAC-TD-n	SPDU	SHORT ACCEPT (not completed) SPDU on T-DATAreq
SACC-TD	SPDU	SHORT ACCEPT CONTINUE SPDU on T-DATAreq
SACC-TC	SPDU	SHORT ACCEPT CONTINUE SPDU in User-data of T-CONrsp
SRFC-TC	SPDU	SHORT REFUSE CONTINUE SPDU in User-data of T-CONrsp
SCNC	SPDU	SHORT CONNECT CONTINUE SPDU
SCN-TC	SPDU	SHORT CONNECT SPDU on User-data of T-CONreq
SCN-TD	SPDU	SHORT CONNECT SPDU on T-DATAreq
SDN	SPDU	SHORT DISCONNECT SPDU
SDT	SPDU	SHORT DATA TRANSFER SPDU
SFN	SPDU	SHORT FINISH SPDU
SRF-TC-nr	SPDU	SHORT REFUSE SPDU (with no reuse of transport connection) on User-data of T-CONrsp
SRF-TC-r	SPDU	SHORT REFUSE SPDU (with reuse of transport connection) on User-data of T-CONrsp
SRF-TD-nr	SPDU	SHORT REFUSE SPDU (with no reuse of transport connection) on T-DATAreq
SRF-TD-r	SPDU	SHORT REFUSE SPDU (with reuse of transport connection) on T-DATAreq

In Table A.3, change the descriptions of TCONreq *and* TCONrsp*:*

TCONreq	TS-provider	T-CONNECT request with no SPDU on the User-data
TCONrsp	TS-provider	T-CONNECT response with no SPDU on the User-data

Add the following entries to Table A.7:

p205	Local choice
p206	SCN SPDU (combined with SCNC if received) is not acceptable to SPM for permanent or transient reasons
p207	Local choice and SHORT CONNECT SPDU can be sent in User-data of T-CONNECT request
p208	Null-encoding protocol option has been selected
p209	SPM can fully interpret the Connection Summary parameter or there is no Connection Summary parameter on the received SPDU
p210	SHORT ACCEPT SPDU can be sent in User-data of T-CONNECT response
p211	SHORT REFUSE SPDU can be sent in User-data of T-CONNECT response

Add to list of Notes to Tables A.8 to A.25:

6 SyABind means generate event SUABind if the User-data field of the T-DISCONNECT indication is one octet with the value 1; generate event SPAPind otherwise.

Add the following entries to Table A.8:

State / Event	STA01 idle No TC	STA01B await TCONcnf	STA01C idle TC con	STA01E await SAC-TC	STA01F await SCONrsp (after SCN-TC rcv)
SCONreq	p207 [2] SCN-TC STA01E ¬p207 TCONreq [2] STA01B		p01 & p204 CN STA02B p01 & p205 SCN-TD STA02C p01 & ¬p204 & ¬p205 CN STA02A		
SACC-TC				STA02C	
SRFC-TC				STA02C	
SCN-TC	p209 & ¬p206 SCONind STA01F ¬p209 & ¬p206 SAC-TC-n STA07 p206 TDISreq STA01				
SCN-TD			¬p01 & ¬p206 & p209 SCONind STA08B ¬p01 & ¬p206 & ¬p209 SAC-TD-n STA07 ¬p01 & p206 & ¬p02 SRF-TD-nr [4] STA16 ¬p01 & p206 & p02 SRF-TD-r STA01C p01 TDISreq STA01		

Additions to Table A.8 (continued):

State / Event	STA01 idle No TC	STA01B await TCONcnf	STA01C idle TC con	STA01E await SAC-TC	STA01F await SCONrsp (after SCN-TC rcv)
SCONrsp+					p208 & p210 SAC-TC-y STA700 p208 & ¬p210 SACC-TC SAC-TD-y STA700 ¬p208 & p210 SAC-TC-y [5] [11] STA713 ¬p208 & ¬p210 SACC-TC SAC-TD-y [5] [11] STA713
SCONrsp–					p02 & p211 SRF-TC-r STA01C p02 & ¬p211 SRFC-TC SRF-TD-r STA01C ¬p02 & p211 SRF-TC-nr [4] STA16 ¬p02 & ¬p211 SRFC-TC SRF-TD-nr [4] STA16
SAC-TC-n				SCNC STA02D	

Additions to Table A.8 (continued):

State / Event	STA01 idle No TC	STA01B await TCONcnf	STA01C idle TC con	STA01E await SAC-TC	STA01F await SCONrsp (after SCN-TC rcv)
SAC-TC-y				p208 SCONcnf+ STA700 ¬p208 SCONcnf+ [5] [11] STA713	
SRF-TC-r				p02 SCONcnf– STA01C ¬p02 SCONcnf– TDISreq STA01	
SRF-TC-nr				SCONcnf– TDISreq STA01	
SRF-TD-nr					
TCONcnf		p204 CN STA02B ¬p204 & p205 SCN-TD STA02C ¬p204 & ¬p205 CN STA02A		p205 CN STA02A ¬p205 SCONcnf— STA01C	

Additions to Table A.8 (concluded):

State / Event	STA02C await SAC-TD	STA02D await SACC	STA07 await SCNC	STA08B await SCONrsp (after SCN-TD rcv)	STA08C await SCONrsp (after SCNCrcv)
SCNC			¬p206 SCONind STA08C p206 & ¬p02 SRF-TD-nr [4] STA16 p206 & p02 SRF-TD-r STA01C		
SCONrsp+				p208 SAC-TD-y STA700 ¬p208 SAC-TD-y [5] [11] STA713	p208 SACC-TD STA700 ¬p208 SACC-TD [5] [11] STA713
SCONrsp–					p02 SRF-TD-r STA01C ¬p02 SRF-TD-nr [4] STA16
SACC-TD		p208 SCONcnf+ STA700 ¬p208 SCONcnf+ [5] [11] STA713 [6]			
SAC-TD-n	SCNC STA02D				
SAC-TD-y	p208 SCONcnf+ STA700 ¬p208 SCONcnf+ [5] [6] [11] STA713				
SRF-TD-r	p02 SCONcnf– STA01C ¬p02 SCONcnf– TDISreq STA01	p02 SCONcnf– STA01C ¬p02 SCONcnf– TDISreq STA01			
SRF-TD-nr	SCONcnf– TDISreq STA01	SCONcnf– TDISreq STA01			

Add, as shown below, to the cell identified by [AC, STA02A] in Table A.8:

p208 SCONcnf+ [5] STA700 ¬p208 SCONcnf+ [5] [11] STA713 [6]

Add, as shown below, to the cell identified by [SCONrsp+, STA08] in Table A.8:

p208 AC [5] STA700 ¬p208 AC [5] [11] STA713 [6]

Add the following entries to Table A.9:

State / Event	STA700 data transfer (null-encoding)
SDTreq	NL STA700
NL	SDTind STA700

In Table A.9, insert a row with Event SDT, and identical entries to row DT.

In Table A.9, for all cells in row SDTreq that include outgoing event DT, add an equivalent action list with SDT and the same final state. The new action list is headed by p211, the old list by ¬p211 (in each case, "anded" with any existing predicates).

In Table A.15, insert a row with Event SDN, and identical entries to row DN.

In Table A.15, insert a row with Event SFN-nr, and identical entries to row FN-nr.

In Table A.15, insert a row with Event SFN-r, and identical entries to row FN-r.

In Table A.15, for all cells in row SRELreq that include outgoing event FN-nr, add an equivalent action list with SFN-nr and the same final state. The new action list is headed by p211, the old list by ¬p211 (in each case, "anded" with any existing predicates).

In Table A.15, for all cells in row SRELreq that include outgoing event FN-n, add an equivalent action list with SFN-n and the same final state. The new action list is headed by p211, the old list by ¬p211 (in each case, "anded" with any existing predicates).

In Table A.15, for all cells in row SRELrsp+ that include outgoing event DN, add an equivalent action list with DN and the same final state. The new action list is headed by p211, the old list by ¬p211 (in each case, "anded" with any existing predicates).

In Table A.16, insert a row with Event SAB-nr, and identical entries to row AB-nr.

In Table A.16, insert a row with Event SAB-r, and identical entries to row AB-r.

In Table A.16, for all cells in row SUABreq that include outgoing event AB-nr, add an equivalent action list with SAB-nr and the same final state. The new action list is headed by p211, the old list by ¬p211 (in each case, "anded" with any existing predicates).

In Table A.16, for all cells in row SUABreq that include outgoing event AB-n, add an equivalent action list with SAB-n and the same final state. The new action list is headed by p211, the old list by ¬p211 (in each case, "anded" with any existing predicates).

In Table A.16 add the columns for the following states:

STA01E, STA01F, STA700

For all these columns, all rows are empty except the following:

SUABreq	TDISreq STA01
TDISind	SPABind STA01

In Table A.16 add the columns for the following states:

STA02C, STA02D, STA07, STA08B

The entries for these columns are identical to those for STA02A.

ICS 35.100.50

Descriptors: data processing, information interchange, network interconnection, open systems interconnection, session layer, communication procedure, control procedures, protocols, implementation.

Price based on 32 pages

INTERNATIONAL STANDARD

ISO/IEC 8327-1

Second edition
1996-09-15

Information technology — Open Systems Interconnection — Connection-oriented Session protocol: Protocol specification

Technologies de l'information — Interconnexion de systèmes ouverts (OSI) — Protocole de session en mode orienté connexion: Spécification du protocole

Reference number
ISO/IEC 8327-1:1996(E)

Contents

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 8327-1 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 21, *Open Systems Interconnection, data management and open distributed processing*, in collaboration with ITU-T. The identical text is published as ITU-T Recommendation X.225.

This second edition cancels and replaces the first edition (ISO 8327:1987), which has been technically revised. It also incorporates Amendment 3:1992.

ISO/IEC 8327 consists of the following parts, under the general title *Information technology — Open Systems Interconnection — Connection-oriented Session protocol*:

— *Part 1: Protocol specification*

— *Part 2: Protocol Implementation Conformance Statement (PICS) proforma*

Annexes A to C form an integral part of this part of ISO/IEC 8327. Annex D is for information only.

Introduction

This Recommendation | International Standard is one of a set of Recommendations | International Standards produced to facilitate the interconnection of computer systems. The set of Recommendations | International Standards covers the services and protocols required to achieve such interconnection.

This Recommendation | International Standard is positioned with respect to other related Recommendations | International Standards by the layers defined in the Reference Model for Open Systems Interconnection (see ITU-T Rec. X.200 | ISO/IEC 7498-1). It is most closely related to and lies within the field of application of the Session Service Definition (see ITU-T Rec. X.215 | ISO/IEC 8326). It also uses and references the Transport Service Definition (see ITU-T Rec. X.214 | ISO/IEC 8072), whose provisions it assumes in order to accomplish the aims of the session protocol. The interrelationship of these Recommendations | International Standards is depicted in Figure Intro. 1.

This Recommendation | International Standard specifies a single protocol with a common encoding.

It is intended that the session protocol should be general enough to cater for the total range of session service users without restricting future extensions.

The protocol is structured so that subsets of protocol can be defined.

The primary aim of this Recommendations | International Standards is to provide a set of rules for communication expressed in terms of the procedures to be carried out by peer session entities at the time of communication. These rules for communication are intended to provide a sound basis for development in order to serve a variety of purposes:

a) as a guide for implementors and designers;

b) for use in the testing and procurement of equipment;

c) as a part of an agreement for the admittance of systems into the open systems environment;

d) as a refinement to the understanding of OSI.

It is expected that the initial users of this Recommendation | International Standard will be designers and implementors of equipment and the Recommendation | International Standard contains, in Notes or in annexes, guidance on the implementation of the procedures defined in this Recommendation | International Standard.

It should be noted that, as the number of valid protocol sequences is very large, it is not possible with current technology to verify that an implementation will operate the protocol defined in this Recommendation | International Standard correctly under all circumstances. It is possible by means of testing to establish confidence that an implementation correctly operates the protocol in a representative sample of circumstances. It is, however, intended that this Recommendation | International Standard can be used in circumstances where two implementations fail to communicate in order to determine whether one or both have failed to operate the protocol correctly.

The variations and options available within this Recommendation | International Standard are essential as they enable a session service to be provided for a wide variety of applications. Thus, a minimally conforming implementation will not be suitable for use in all possible circumstances. It is important, therefore, to qualify all references to this Recommendation | International Standard with statements of the options provided or required or with statements of the intended purpose of provision or use.

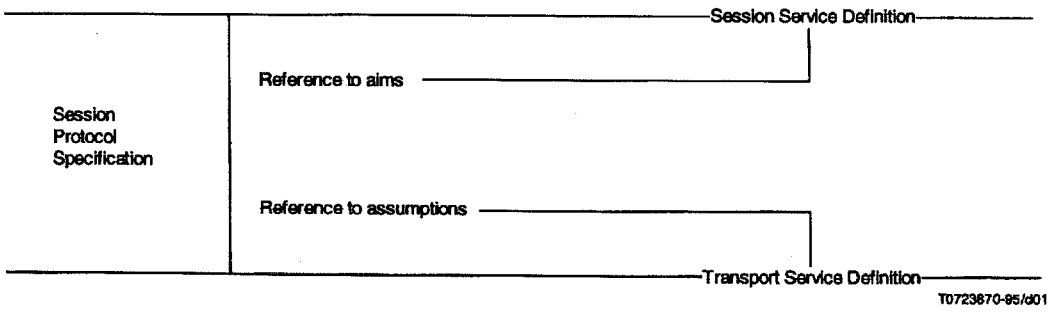

Figure Intro. 1 – Relationship between the connectionless session protocol and adjacent services

INTERNATIONAL STANDARD

ITU-T RECOMMENDATION

INFORMATION TECHNOLOGY – OPEN SYSTEMS INTERCONNECTION – CONNECTION-ORIENTED SESSION PROTOCOL: PROTOCOL SPECIFICATION

SECTION 1 – GENERAL

1 Scope

This Recommendation | International Standard specifies:

a) procedures for a single protocol for the transfer of data and control information from one session entity to a peer session entity;

b) the means of selecting the functional units to be used by the session entities;

c) the structure and encoding of the session protocol data units used for the transfer of data and control information;

The procedures are defined in terms of:

a) the interactions between peer session entities through the exchange of session protocol data units;

b) the interactions between a session entity and the session service user in the same system through the exchange of session service primitives;

c) the interactions between a session entity and the transport service provider through the exchange of transport service primitives.

These procedures are applicable to instances of communication between systems which support the session layer of the OSI Reference Model and which wish to interconnect in an open systems environment.

This Recommendation | International Standard also specifies conformance requirements for systems implementing these procedures. It does not contain tests which can be used to demonstrate this conformance.

The text pertaining to the symmetric synchronization functional unit is not applicable to the support of ITU-T applications.

2 Normative references

The following Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation | International Standard. At the time of publication, the editions indicated were valid. All Recommendations and International Standards are subject to revision, and parties to agreements based on this Recommendation | International Standard are encouraged to investigate the possibility of applying the most recent edition of the Recommendations and International Standards listed below. Members of ISO and IEC maintain registers of currently valid International Standards. The Telecommunication Standardization Bureau of the ITU maintains a list of currently valid ITU-T Recommendations.

2.1 Identical Recommendations | International Standards

- ITU-T Recommendation X.200 (1994) | ISO/IEC 7498-1:1994, *Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model.*
- ITU-T Recommendation X.214 (1993) | ISO/IEC 8072:1994, *Information technology – Open Systems Interconnection – Transport service definition.*

- ITU-T Recommendation X.215 (1995) | ISO/IEC 8326:1996, *Information technology – Open Systems Interconnection – Session service definition.*
- ITU-T Recommendation X.245 (1995) | ISO/IEC 8327-2:[1], *Information technology – Open Systems Interconnection – Connection-oriented Session protocol: Protocol Implementation Conformance Statement (PICS) proforma.*

2.2 Paired Recommendations | International Standards equivalent in technical content

- CCITT Recommendation X.290 (1992), *OSI conformance testing methodology and framework for protocol Recommendations for CCITT applications – General concepts.*

  ISO/IEC 9646-1:1994, *Information technology – Open Systems Interconnection – Conformance testing methodology and framework – Part 1: General concepts.*
- CCITT Recommendation X.291 (1992), *OSI conformance testing methodology and framework for protocol Recommendations for CCITT applications – Abstract test suite specification.*

  ISO/IEC 9646-2:1994, *Information technology – Open Systems Interconnection – Conformance testing methodology and framework – Part 2: Abstract Test Suite specification.*

2.3 Additional References

- ITU-T Recommendation T.62 (1993), *Control procedures for teletex and Group 4 facsimile services.*

  NOTE – ITU-T Recommendation T.62 is not essential for the application of this Recommendation | International Standard, but is included in the list of references as it has been referred to, for information, in relation to interworking with the ITU-T Telematic services (see Annexes B and C).

3 Definitions

For the purposes of this Recommendation | International Standard, the following definitions apply.

NOTE – The definitions contained in this clause make use of abbreviations defined in clause 4.

3.1 This Recommendation | International Standard is based on the concepts developed in ITU-T Rec. X.200 | ISO 7498-1, and makes use of the following terms defined in it:

a) expedited session service data unit;
b) session connection;
c) Session Layer;
d) session protocol data unit;
e) session service;
f) session service access point;
g) session service data unit;
h) Transport Layer;
i) transport connection;
j) transport service;
k) transport service access point;
l) concatenation;
m) segmenting.

3.2 This Recommendation | International Standard is also based on concepts developed in ITU-T Rec. X.215 | ISO/IEC 8326 and makes use of the following terms defined in it:

a) token;
b) calling SS-user;
c) called SS-user;
d) sending SS-user;

[1] To be published.

e) receiving SS-user;

f) requesting SS-user;

g) accepting SS-user;

h) requestor;

i) acceptor.

NOTE – The following terms used in this Recommendation | International Standard are used in relation to tokens and are explained in ITU-T Rec. X.215 | ISO/IEC 8326:

a) assigned;

b) not assigned;

c) available;

d) not available.

3.3 This Recommendation | International Standard also uses concepts developed in CCITT Rec. X.290 | ISO/IEC 9646-1 and makes use of the following terms defined in it:

a) Protocol Implementation Conformance Statement (PICS);

b) PICS proforma.

3.4 Session protocol definitions

3.4.1 Session Protocol Machine (SPM): An abstract machine that carries out the procedures specified in this protocol.

NOTE – A session entity is comprised of one or more SPMs.

3.4.2 session service user (SS-user): An abstract representation of the totality of those entities within a single system that make use of the Session Service.

3.4.3 transport service provider (TS-provider): An abstract machine which models the totality of the entities providing the transport service, as viewed by a session entity.

3.4.4 local matter: A decision made by a system concerning its behaviour in the Session Layer that is not subject to the requirements of this protocol.

3.4.5 initiator: An SPM that initiates a CONNECT SPDU.

3.4.6 responder: An SPM with whom an initiator wishes to establish a session connection.

NOTE – Initiator and responder are defined with respect to a single session connection.

3.4.7 sending SPM: An SPM that sends a given SPDU.

3.4.8 receiving SPM: An SPM that receives a given SPDU.

3.4.9 owner (of a token): The SPM to whom a token is assigned.

3.4.10 proposed parameter: The value for a parameter proposed by an SPM in a CONNECT SPDU or an ACCEPT SPDU that it wishes to use on the session connection.

3.4.11 negotiation: The process by which two SPMs agree on a common set of functional units and protocol values and on the initial setting of available tokens.

3.4.12 selected parameter: The value for a parameter that has been chosen for use on the session connection.

3.4.13 valid SPDU: An SPDU which complies with the requirements of ITU-T Rec. X.225 | ISO/IEC 8327-1 with respect to structure and encoding.

3.4.14 invalid SPDU: An SPDU which does not comply with the requirements of ITU-T Rec. X.225 | ISO/IEC 8327-1 with respect to structure and encoding.

3.4.15 protocol error: Use of an SPDU that does not comply with the procedures agreed for the session connection.

3.4.16 transparent (data): SS-user data which is transferred intact between SPMs and which is unavailable for use by the SPMs.

3.4.17 SPDU identifier (SI): Heading information that identifies the SPDU concerned.

3.4.18 length indicator (LI): An indicator that represents the length of an associated parameter field.

3.4.19 parameter field: A group of one or more octets used to represent a particular set of information.

3.4.20 parameter identifier (PI): An identifier, defined in ITU-T Rec. X.225 | ISO/IEC 8327-1, that indicates the type of information contained in its associated parameter field.

3.4.21 PI unit: An element of an SPDU that contains a PI field together with its associated LI field and parameter field.

3.4.22 parameter group identifier (PGI): An identifier, defined in ITU-T Rec. X.225 | ISO/IEC 8327-1, that indicates the type of information contained in its associated parameter field. The associated parameter field may consist of a set of PI units.

3.4.23 PGI unit: An element of an SPDU that contains a PGI field together with its associated LI field and parameter field.

3.4.24 parameter value (PV): Information that represents the value of the parameter identified by either a PI or PGI.

3.4.25 local variable: A local variable within the SPM which is used as a means of clarifying the effects of certain actions and clarifying the conditions under which certain actions are permitted.

4 Abbreviations

For the purposes of this Recommendation | International Standard, the following abbreviations apply.

4.1 Data units

SPDU session-protocol-data-unit

SSDU session-service-data-unit

TSDU transport-service-data-unit

4.2 SPDU fields

SI SPDU identifier (see 3.4.17)

LI Length indicator (see 3.4.18)

PI Parameter identifier (see 3.4.20)

PGI Parameter group identifier (see 3.4.22)

PV Parameter value (see 3.4.24)

4.3 Timer variables

TIM Disconnection and abort timer

4.4 Miscellaneous

PICS Protocol Implementation Conformance Statement

SPM Session Protocol Machine (see 3.4.1)

SS session-service

SSAP session-service-access-point

TSAP transport-service-access-point

4.5 Local variables

Vact	See 5.9.1
Vnextact	See 5.9.2
V(A)	See 5.9.3.1
V(M)	See 5.9.3.2
V(R)	See 5.9.3.3
Vsc	See 5.9.3.4 and 5.9.4.4
V(Ado)	See 5.9.3.5
V(Adi)	See 5.9.3.6
V(As), V(Ar)	See 5.9.4.1
V(Ms), V(Mr)	See 5.9.4.2
V(Rs), V(Rr)	See 5.9.4.3

5 Overview of the session protocol

5.1 Model of the session layer

The SPM (see the Note) within the session layer communicates with the SS-user through an SSAP by means of the service primitives as defined by the session service definition ITU-T Rec. X.215 | ISO/IEC 8326. Service primitives will cause or be the result of session protocol data unit exchanges between the peer SPMs using a transport connection. These protocol exchanges are effected using the services of the transport layer as defined by the transport service definition ITU-T Rec. X.214 | ISO/IEC 8072 through two TSAPs.

Session connection endpoints are identified in end systems by an internal, implementation dependent, mechanism so that the SS-user and the SPM can refer to each session connection.

The model of the session layer is illustrated in Figure 1.

NOTE – A session entity is comprised of one or more SPMs.

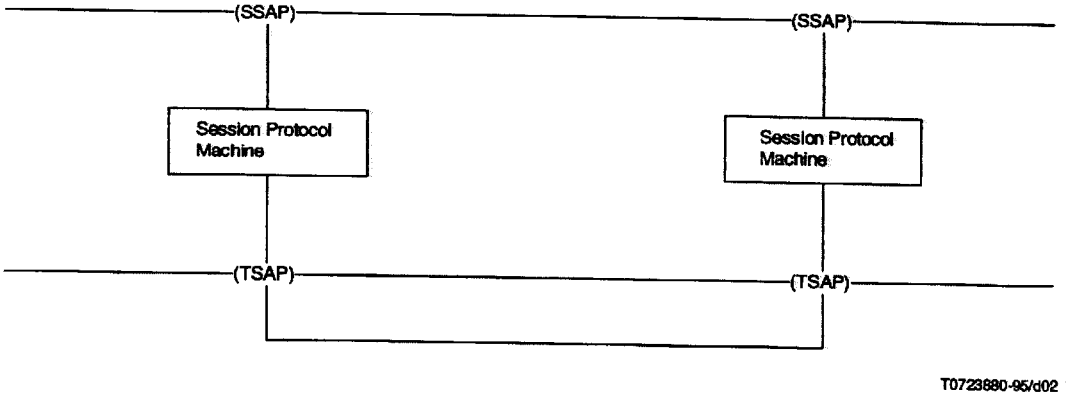

Figure 1 – Model for the session layer

5.2 Services provided by the session layer

The protocol specified in this Recommendation | International Standard supports the session service defined in ITU-T Rec. X.215 | ISO/IEC 8326. Information is transferred to and from the SS-user using the session service primitives listed in *Table 1*. *Table 1* also defines the SPDUs associated with each of the service primitives.

Table 1 – Session service primitives

Service	Primitives	Associated SPDUs
Session Connection	S-CONNECT request S-CONNECT indication S-CONNECT (accept) response S-CONNECT (accept) confirm S-CONNECT (reject) response S-CONNECT (reject) confirm	CONNECT SPDU CONNECT SPDU ACCEPT SPDU ACCEPT SPDU REFUSE SPDU REFUSE SPDU
Normal Data Transfer	S-DATA request S-DATA indication	DATA TRANSFER SPDU DATA TRANSFER SPDU
Expedited Data Transfer	S-EXPEDITED-DATA request S-EXPEDITED-DATA indication	EXPEDITED DATA SPDU EXPEDITED DATA SPDU
Typed Data Transfer	S-TYPED-DATA request S-TYPED-DATA indication	TYPED DATA SPDU TYPED DATA SPDU
Capability Data Exchange	S-CAPABILITY-DATA request S-CAPABILITY-DATA indication S-CAPABILITY-DATA response S-CAPABILITY-DATA confirm	CAPABILITY DATA SPDU CAPABILITY DATA SPDU CAPABILITY DATA ACK SPDU CAPABILITY DATA ACK SPDU
Give Tokens	S-TOKEN-GIVE request S-TOKEN-GIVE indication	GIVE TOKENS SPDU GIVE TOKENS SPDU
Please Tokens	S-TOKEN-PLEASE request S-TOKEN-PLEASE indication	PLEASE TOKENS SPDU PLEASE TOKENS SPDU
Give Control	S-CONTROL-GIVE request S-CONTROL-GIVE indication	GIVE TOKENS CONFIRM SPDU GIVE TOKENS CONFIRM SPDU
Minor Synchronization Point	S-SYNC-MINOR request S-SYNC-MINOR indication S-SYNC-MINOR response S-SYNC-MINOR confirm	MINOR SYNC POINT SPDU MINOR SYNC POINT SPDU MINOR SYNC ACK SPDU MINOR SYNC ACK SPDU
Symmetric Synchronize	S-SYNC-MINOR request S-SYNC-MINOR indication S-SYNC-MINOR response S-SYNC-MINOR confirm	MINOR SYNC POINT SPDU MINOR SYNC POINT SPDU MINOR SYNC ACK SPDU MINOR SYNC ACK SPDU
Major Synchronization Point	S-SYNC-MAJOR request S-SYNC-MAJOR indication S-SYNC-MAJOR response S-SYNC-MAJOR confirm	MAJOR SYNC POINT SPDU MAJOR SYNC POINT SPDU MAJOR SYNC ACK SPDU MAJOR SYNC ACK SPDU
Resynchronize	S-RESYNCHRONIZE request S-RESYNCHRONIZE indication S-RESYNCHRONIZE response S-RESYNCHRONIZE confirm	RESYNCHRONIZE SPDU RESYNCHRONIZE SPDU RESYNCHRONIZE ACK SPDU RESYNCHRONIZE ACK SPDU
P-Exception Report	S-P-EXCEPTION-REPORT indication	EXCEPTION REPORT SPDU
U-Exception Reporting	S-U-EXCEPTION-REPORT request S-U-EXCEPTION-REPORT indication	EXCEPTION DATA SPDU EXCEPTION DATA SPDU
Activity Start	S-ACTIVITY-START request S-ACTIVITY-START indication	ACTIVITY START SPDU ACTIVITY START SPDU
Activity Resume	S-ACTIVITY-RESUME request S-ACTIVITY-RESUME indication	ACTIVITY RESUME SPDU ACTIVITY RESUME SPDU

Table 1 – Session service primitives (*concluded*)

Service	Primitives	Associated SPDUs
Activity Interrupt	S-ACTIVITY-INTERRUPT request S-ACTIVITY-INTERRUPT indication S-ACTIVITY-INTERRUPT response S-ACTIVITY-INTERRUPT confirm	ACTIVITY INTERRUPT SPDU ACTIVITY INTERRUPT SPDU ACTIVITY INTERRUPT ACK SPDU ACTIVITY INTERRUPT ACK SPDU
Activity Discard	S-ACTIVITY-DISCARD request S-ACTIVITY-DISCARD indication S-ACTIVITY-DISCARD response S-ACTIVITY-DISCARD confirm	ACTIVITY DISCARD SPDU ACTIVITY DISCARD SPDU ACTIVITY DISCARD ACK SPDU ACTIVITY DISCARD ACK SPDU
Activity End	S-ACTIVITY-END request S-ACTIVITY-END indication S-ACTIVITY-END response S-ACTIVITY-END confirm	ACTIVITY END SPDU ACTIVITY END SPDU ACTIVITY END ACK SPDU ACTIVITY END ACK SPDU
Orderly Release	S-RELEASE request S-RELEASE indication S-RELEASE (accept) response S-RELEASE (accept) confirm S-RELEASE (reject) response S-RELEASE (reject) confirm	FINISH SPDU FINISH SPDU DISCONNECT SPDU DISCONNECT SPDU NOT FINISHED SPDU NOT FINISHED SPDU
U-Abort	S-U-ABORT request S-U-ABORT indication	ABORT SPDU ABORT SPDU
P-Abort	S-P-ABORT indication	ABORT SPDU

5.3 Services assumed from the transport layer

The protocol specified in this Recommendation | International Standard assumes the use of the connection-oriented transport service defined in ITU-T Rec. X.214 | ISO/IEC 8072.

Information is transferred to and from the TS provider in the transport service primitives listed in Table 2.

5.4 Functions of the session layer

5.4.1 Overview of functions

The functions in the Session Layer are those necessary to bridge the gap between the services available from the Transport Layer and those offered to the SS-users.

The functions in the Session Layer are concerned with dialogue management, data flow synchronization, and data flow resynchronization.

These functions are described below; the descriptions are grouped into those concerned with the connection establishment phase, the data transfer phase, and the release phase.

5.4.2 Connection establishment phase

The purpose of the connection establishment phase is to establish a session connection between two SS-users, and:

a) to map session addresses onto transport addresses;

b) to select transport quality of service parameters needed (see 6.1.4);

c) to negotiate session parameters (see 7.1, 7.2 and 7.4);

d) to transfer session selectors (see 7.1 and 7.4) if required;

e) to distinguish between session connections (see 7.1 and 7.4);

f) to transfer transparent user data (see 7.1, 7.2, 7.3 and 7.4);

g) to select a protocol version (see the Note).

NOTE – This Recommendation | International Standard specifies the following protocol versions:

a) Protocol Version 1 which imposes restrictions on the length of the user data field;

b) Protocol Version 2 which imposes no explicit restrictions on the length of the user data field.

Annex D identifies the differences between Protocol Version 1 and Protocol Version 2.

Table 2 – Transport service primitives

Primitive	X/Y	Parameters
T-CONNECT request T-CONNECT indication	X	Called Address Calling Address Expedited Data option Quality of Service TS-user-data
T-CONNECT response T-CONNECT confirm	X	Quality of Service Responding Address Expedited Data option TS-user-data
T-DATA request T-DATA indication	X	TS-user-data
T-EXPEDITED-DATA request T-EXPEDITED-DATA indication	Y	TS-user-data
T-DISCONNECT request	X	TS-user-data
T-DISCONNECT indication	X	Disconnect reason TS-user-data
X The session protocol assumes that this service is always available. Y The session protocol assumes that this service is provided by the transport layer when requested by the SPM during the session connection establishment phase.		

5.4.3 Data transfer phase

The purpose of the data transfer phase is to transport SSDUs between two SS-users connected by a session connection. This purpose is achieved by means of transmission of SPDUs and by the following functions, each of which may or may not be used, depending on the functional units selected in the session connection establishment phase. These concepts are defined in ITU-T Rec. X.215 | ISO/IEC 8326:

a) Normal data transfer (see 7.11), which may involve segmenting of SSDUs into SPDUs and reassembly by the destination SPM; and concatenation and separation of certain SPDUs. There are two modes of operation.

1) half-duplex, when the right to send data is restricted to the owner of the data token;

2) duplex, when there is no restriction on the right to send data.

b) Token management (see 7.16 to 7.19), to enable the SS-users to request and transfer tokens which control the exclusive right to exercise certain functions (see Table 5).

c) Exception reporting (see 7.27 and 7.28), to enable the SS-provider or the SS-user to report exception conditions that are less severe than those requiring abort.

d) Typed data transfer (see 7.13), to enable transfer of information which is not subject to assignment of the data token.

e) Minor synchronization point (see 7.20 and 7.21), to enable the SS-users to define minor synchronization points in the normal data flow. These minor synchronization points may optionally be confirmed, but have no implications on the data flow. Minor synchronization points are identified by synchronization point serial numbers. The serial number is incremented by one on each occasion that a minor synchronization point is placed in the data flow, and each time a minor synchronization point is received, such that both SS-users have the same serial numbers for the same synchronization point.

f) Symmetric synchronization (see 7.18 to 7.23), to enable the SS-users to define synchronization points independently on their sending flows. These synchronization points may optionally be confirmed, but have no implications on the data flow. Two serial numbers are provided: one to identify points on each direction of data flow. The appropriate serial number is incremented each time a synchronization point is sent or received. The ability to define a minor symmetric synchronization point is not token controlled.

g) Data separation is used in association with either minor synchronization or symmetric synchronization. This allows the user to set specific minor synchronization points which have the property of protecting the flow sent before these points from being purged by a subsequent resynchronize. When such a point is set in the data flow, SPDUs received by the requesting SPM before this point has been acknowledged will not be discarded by a subsequent resynchronize.

h) Major synchronization point [see 7.22 and 7.23 and e) above], to enable the SS-users to define major synchronization points in the normal data flow. These major synchronization points are required to be confirmed before the requesting SS-user is permitted to send any subsequent data on either the normal flow or the expedited flow and as such clearly separate the dialogue units.

i) Resynchronize (see 7.24 and 7.25), a function that allows a session connection to be set or reset to a defined synchronization point and reassign the tokens.

j) Expedited data transfer (see 7.12), a function used to convey a limited amount of user data with special handling. Such data may bypass normal data en route, but will be delivered prior to any data subsequently sent on the transport normal flow or the transport expedited flow.

k) Activity management (see 7.29 to 7.36) provides a means explicitly to start, end, resume, interrupt or discard an activity. This provides a way:

   1) to identify the entered activity and commence synchronization point serial numbering;

   2) to identify the continued activity and reset the synchronization point serial number in case of resumption.

l) Capability data exchange (see 7.14 and 7.15), to provide a confirmed transfer of SS-user data.

5.4.4 Connection release phase

The purpose of the release phase is to provide disconnection of the session connection, by using the following functions:

a) orderly release (negotiated and non-negotiated);

b) abort (provider and user initiated);

c) transfer of transparent user data.

5.5 Protocol version numbers

Protocol Version 1 of the session protocol is a subset of Protocol Version 2. Both protocol versions are both specified in this edition of this Recommendation | International Standard (see the Note).

Protocol Version 1 imposes restrictions on the length of the user data field. Protocol Version 2 imposes no explicit restrictions on the length of the user data field.

The restrictions defining Protocol Version 1 are specified in clauses 6, 7 and 8. Annex D lists, for information, the differences between the two protocol versions.

NOTE – Edition 1 of this Recommendation | International Standard, (Rec.X.225 (1984) | ISO 8327-1: 1987), defined only Protocol Version 1 of the session protocol.

5.6 Functional units

Functional units are logical groupings of related elements of procedure defined by this Recommendation | International Standard for the purpose of:

a) negotiation for use during session connection establishment;

b) specification of conformance requirements.

The SPDUs associated with elements of procedure for each functional unit are specified in Table 3. This table does not take account of additional SPDUs (GT SPDU and PT SPDU) required by the concatenation rules (see 6.3.7).

Tokens are associated with functional units (see 5.7).

5.6.1 Kernel functional unit

The kernel functional unit supports the basic protocol elements of procedure required to establish a session connection, transfer normal data and release the session connection.

5.6.2 Negotiated release functional unit

The negotiated release functional unit supports the negotiated release service which enables the SS-users to negotiate the orderly release of the session connection. If this functional unit has been selected, an attempt to release the session connection may be refused by the accepting SS-user.

5.6.3 Half-duplex functional unit

The half-duplex functional unit is used to control the right to send data. It is not valid to select both this functional unit and the duplex functional unit for use on the same session connection.

5.6.4 Duplex functional unit

The duplex functional unit is used when the right to send data is not controlled. It is not valid to select both this functional unit and the half-duplex functional unit for use on the same session connection.

5.6.5 Expedited data functional unit

The expedited data functional unit supports the expedited data service and allows the transfer of a limited amount of SS-user data.

The services supported by this functional unit can only be requested when the transport expedited flow is available to this session connection.

5.6.6 Typed data functional unit

The typed data functional unit enables the SS-users to transfer data in a manner which is not subject to the control imposed by the availability of the data token.

5.6.7 Capability data exchange functional unit

The capability data functional unit supports the capability data exchange service, which allows a confirmed transfer of SS-user data when the activity management functional unit has been selected, but when no activity is in progress.

5.6.8 Minor synchronize functional unit

The minor synchronize functional unit supports the minor synchronization service which enables the SS-user to request that the SPM places minor synchronization points in the normal data flow. These minor synchronization points are identified by serial numbers. It is not valid to select both this functional unit and the symmetric synchronize functional unit for use on the same session connection.

5.6.9 Symmetric synchronize functional unit

The symmetric synchronize functional unit supports the symmetric synchronize service which enables SS-users to independently define minor synchronization points in their respective sending flows. These sync points are identified by serial numbers; two serial numbers are provided, each representing points in one direction of flow. The synchronize-minor token is not available. It is not valid to select both this functional unit and the minor synchronize functional unit for use on the same session connection.

5.6.10 Data separation functional unit

The data separation functional unit is always associated with either the minor synchronize functional unit or the symmetric synchronize functional unit. It allows the user to define minor synchronization points which clearly separate the normal flow before and after these points and which are protected against possible discard by a subsequent resynchronize. It is not valid to select both this functional unit and the activity management functional unit for use on the same session connection.

Table 3 – Functional units

Functional unit	SPDU code	SPDU name	Reference
Kernel	CN	CONNECT (see Note 1)	7.1
	OA	OVERFLOW ACCEPT (see Note 2)	7.2
	CDO	CONNECT DATA OVERFLOW (see Note 2)	7.3
	AC	ACCEPT (see Note 1)	7.4
	RF	REFUSE (see Note 1)	7.5
	FN	FINISH	7.6
	DN	DISCONNECT	7.7
	AB	ABORT	7.9
	AA	ABORT ACCEPT (see Note 3)	7.10
	DT	DATA TRANSFER	7.11
	PR	PREPARE (see Note 7)	7.26
Negotiated release	NF	NOT FINISHED	7.8
	GT	GIVE TOKENS (see Note 5)	7.16
	PT	PLEASE TOKENS (see Note 5)	7.17
Half-duplex	GT	GIVE TOKENS (see Note 4)	7.16
	PT	PLEASE TOKENS (see Note 4)	7.17
Duplex		No additional associated SPDUs	
Expedited data	EX	EXPEDITED DATA	7.12
Typed data	TD	TYPED DATA	7.13
Capability data exchange	CD	CAPABILITY DATA	7.14
	CDA	CAPABILITY DATA ACK	7.15
Minor synchronize	MIP	MINOR SYNC POINT	7.20
	MIA	MINOR SYNC ACK	7.21
	GT	GIVE TOKENS (see Note 6)	7.16
	PT	PLEASE TOKENS (see Note 6)	7.17
Symmetric synchronize	MIP	MINOR SYNC POINT	7.20
	MIA	MINOR SYNC ACK	7.21
Data separation		No additional associated SPDUs	
Major synchronize	MAP	MAJOR SYNC POINT	7.22
	MAA	MAJOR SYNC ACK	7.23
	PR	PREPARE (see Note 7)	7.26
	GT	GIVE TOKENS (see Note 8)	7.16
	PT	PLEASE TOKENS (see Note 8)	7.17
Resynchronize	RS	RESYNCHRONIZE	7.24
	RA	RESYNCHRONIZE ACK	7.25
	PR	PREPARE (see Note 7)	7.26
Exceptions	ER	EXCEPTION REPORT	7.27
	ED	EXCEPTION DATA	7.28
Activity management	AS	ACTIVITY START	7.29
	AR	ACTIVITY RESUME	7.30
	AI	ACTIVITY INTERRUPT	7.31
	AIA	ACTIVITY INTERRUPT ACK	7.32
	AD	ACTIVITY DISCARD	7.33
	ADA	ACTIVITY DISCARD ACK	7.34
	AE	ACTIVITY END	7.35
	AEA	ACTIVITY END ACK	7.36
	PR	PREPARE (see Note 7)	7.26
	GT	GIVE TOKENS (see Note 8)	7.16
	PT	PLEASE TOKENS (see Note 8)	7.17
	GTC	GIVE TOKENS CONFIRM (see Note 9)	7.18
	GTA	GIVE TOKENS ACK (see Note 9)	7.19

Table 3 – Functional units (*concluded*)

NOTES

1 An implementation (see clause 9) is required to be able to:
   a) send a CONNECT SPDU and receive an ACCEPT SPDU or a REFUSE SPDU
   b) receive a CONNECT SPDU and send an ACCEPT SPDU or a REFUSE SPDU
   c) send and receive both.
2 These SPDUs are only used when the SSDU passed in the S-CONNECT request is segmented [see 6.3.5 b)].
3 Reception and correct action is mandatory; transmission is optional if the transport connection is not to be reused (see 7.10.2).
4 Used to manage the data token.
5 Used to manage the release token.
6 Used to manage the synchronize-minor token.
7 PREPARE SPDU is mandatory if the transport expedited flow is available to this session connection.
8 Used to manage the major/activity token.
9 Used only on session connections on which activity management has been selected.

5.6.11 Major synchronize functional unit

The major synchronize functional unit supports the major synchronize service which enables the SS-user to request that the SPM places major synchronization points in the normal data flow. These major synchronization points are identified by serial numbers, and clearly separate the data flow before and after the major synchronization point.

5.6.12 Resynchronize functional unit

The resynchronize functional unit supports the resynchronize service which enables the SS-users to modify the synchronization point serial number and reassign the tokens.

5.6.13 Exceptions functional unit

The exceptions functional unit allows both the SPM and the SS-users to report detected errors, rather than aborting the session connection.

This functional unit can only be selected when the half-duplex functional unit has been selected.

5.6.14 Activity management functional unit

The activity management functional unit supports the activity management services which allow the SS-users to manage synchronized logical pieces of work.

5.7 Tokens

Table 4 specifies those functional units that have tokens associated with them.

The SPM may only send an SPDU listed in Table 5 (and accept the associated service primitive) subject to the availability and assignment of tokens defined in that table.

Table 4 – Tokens associated with functional units

Functional unit	Token
Negotiated release	release token
Half-duplex	data token
Minor synchronize	synchronize-minor token
Major synchronize	major/activity token
Activity management	major/activity token

Table 5 – Token restrictions

SPDUs	data token	synchronize minor token	major/activity token	release token
FINISH SPDU	2	2	2	2
NOT FINISHED SPDU	nr	nr	nr	0
DATA TRANSFER SPDU (half-duplex)	1	nr	nr	nr
DATA TRANSFER SPDU (duplex)	3	nr	nr	nr
CAPABILITY DATA SPDU	2	2	1	nr
GIVE TOKEN SPDU				
(data token)	1	nr	nr	nr
(synchronize-minor token)	nr	1	nr	nr
(major/activity token)	nr	nr	1	nr
(release token)	nr	nr	nr	1
PLEASE TOKEN SPDU				
(data token)	0	nr	nr	nr
(synchronize-minor token)	nr	0	nr	nr
(major/activity token)	nr	nr	0	nr
(release token)	nr	nr	nr	0
GIVE TOKENS CONFIRM SPDU	2	2	1	2
MINOR SYNC POINT SPDU (minor synchronize)	2	1	nr	nr
MINOR SYNC POINT SPDU (symmetric synchronize)	nr	3	nr	nr
MAJOR SYNC POINT SPDU	2	2	1	nr
EXCEPTION REPORT SPDU	0	nr	nr	nr
EXCEPTION DATA SPDU	0	nr	nr	nr
ACTIVITY START SPDU	2	2	1	nr
ACTIVITY RESUME SPDU	2	2	1	nr
ACTIVITY INTERRUPT SPDU	nr	nr	1	nr
ACTIVITY DISCARD SPDU	nr	nr	1	nr
ACTIVITY END SPDU	2	2	1	nr

0 Token available and not assigned to the SS-user who initiated the associated service primitive.
1 Token available and assigned to the SS-user who initiated the associated service primitive.
2 Token not available or token assigned to the SS-user who initiated the associated service primitive.
3 Token not available.
nr No restriction.

5.8 Negotiation

Negotiation takes place between both SPMs during session connection establishment according to the following rules.

5.8.1 Negotiation of functional units

Each SPM proposes use or non-use of each functional unit, except for the kernel functional unit, based on requirements from the SS-users. The functional unit is selected only if both the initiator and the responder propose use of the functional unit.

The capability data functional unit can only be proposed if the activity management functional unit is also proposed.

The exceptions functional unit can only be proposed if the half-duplex functional unit is also proposed.

The data separation functional unit can only be proposed if the minor synchronize functional unit or the symmetric synchronize functional unit is proposed.

5.8.2 Negotiation of initial token settings

When the initiator proposes use of a functional unit that requires a token, it also proposes the initial token setting:

a) initiator's side;

b) responder's side;

c) called SS-user's choice.

If use of the functional unit is selected, the token is set to the side proposed by the initiator. If the initiator proposed "called SS-user choice", the responder's proposed token setting is selected.

5.8.3 Negotiation of initial serial number

When the initiator proposes any of the minor synchronize, symmetric synchronize, major synchronize or resynchronize functional units but does not propose the activity functional unit, it also proposes values for the initial serial number(s). Two initial serial numbers are proposed if the symmetric synchronize functional unit is proposed; otherwise one initial serial number is proposed.

When the initiator proposes any of the minor synchronize, symmetric synchronize, major synchronize, or resynchronize functional units and also proposes the activity management functional unit, it may also propose initial serial number(s).

In all other cases, the initiator does not propose an initial serial number.

When the responder proposes any of the minor synchronize, symmetric synchronize, major synchronize or resynchronize functional units but does not propose the activity management functional unit, it also proposes an initial serial number, which is the first serial number to be used. Two initial serial numbers are proposed if the symmetric synchronize functional unit is proposed; otherwise one initial serial number is proposed.

In all other cases, the responder does not propose an initial serial number.

5.8.4 Negotiation of upper limit serial number

When the initiator proposes an initial serial number (see 5.8.3), it may also propose an upper limit serial number. When the responder proposes an initial serial number (see 5.8.3), it may also propose an upper limit serial number. The responders proposal is used to define the upper limit of the serial number.

The means by which the upper limit serial number is passed between the SS-user and its SPM is a local matter.

5.8.5 Negotiation of protocol version number

Each SPM indicates all the appropriate versions of the protocol that it is capable of supporting. The highest common version number is used.

5.8.6 Negotiation of maximum TSDU size

Each SPM proposes a maximum TSDU size that the initiator is permitted to send. The lesser of the two numbers is used. A zero value is interpreted to mean unlimited TSDU size. If either SPM proposes zero, the initiator may not send segmented data or typed data SSDUs on the session connection.

Each SPM also proposes a maximum TSDU size that the responder is permitted to send. The lesser of the two numbers is used. A zero value is interpreted to mean unlimited TSDU size. If either SPM proposes zero, the responder may not send segmented data or typed data SSDUs on the session connection.

5.9 Local variables

This Recommendation | International Standard uses local variables as a means of clarifying the effect of certain actions and clarifying the conditions under which certain actions are valid.

5.9.1 Vact

Vact is used by the SPM to determine if an activity is in progress when the activity management functional unit has been selected:

- Vact = true: an activity is in progress;
- Vact = false: no activity is in progress.

5.9.2 Vnextact

Vnextact is used by the SPM when the activity management functional unit has been selected:

- Vnextact = true: a MAJOR SYNC POINT SPDU has been sent or received;
- Vnextact = false: an ACTIVITY END SPDU has been sent or received.

5.9.3 Synchronization point variables for use without the symmetric synchronize functional unit

This subclause defines the synchronization variables used by the SPM when the symmetric synchronize functional unit has not been selected for use on the session connection.

5.9.3.1 V(A)

When the symmetric synchronize functional unit has not been selected, V(A) is used by the SPM and is the lowest serial number to which a synchronization point confirmation is expected. No confirmation is expected when V(A) = V(M).

5.9.3.2 V(M)

When the symmetric synchronize functional unit has not been selected, V(M) is used by the SPM and is the next serial number to be used.

5.9.3.3 V(R)

When the symmetric synchronize functional unit has not been selected, V(R) is used by the SPM and is the lowest serial number to which resynchronization restart is permitted.

5.9.3.4 Vsc

When the symmetric synchronize functional unit has not been selected, Vsc is used by the SPM to determine whether or not the SS-user has the right to send minor synchronization point responses. Vsc has the following values:

- Vsc = true: the SS-user has the right to issue minor synchronization point responses when V(A) is less than V(M);
- Vsc = false: the SS-user does not have the right to issue minor synchronization point responses.

NOTE – The manipulation of V(A), V(M), V(R) and Vsc and the circumstances under which they are updated are specified in clause 7 and are summarized in Table A.4.

5.9.3.5 V(Ado)

V(Ado) is used by the SPM and is the highest synchronization point serial number which was sent in a MINOR SYNCHRONIZATION POINT SPDU with the data separation parameter set to true. Received SPDUs cannot be discarded in case of resynchronization when V(Ado) is greater than or equal to V(A).

5.9.3.6 V(Adi)

V(Adi) is used by the SPM and is the highest synchronization point serial number which was received in a MINOR SYNCHRONIZATION POINT SPDU with the data separation parameter set to true.

5.9.4 Synchronization point variables for use with symmetric synchronize functional unit

This subclause defines the synchronization variables used by the SPM when the symmetric synchronize functional unit has been selected for use on the session connection.

The synchronization variables V(A), V(M), and V(R) are each dual number variables (an ordered pair) when the symmetric synchronize functional unit has been selected.

5.9.4.1 V(As), V(Ar)

V(As) and V(Ar) are used by the SPM to manage synchronization point confirmations.

V(As) is the lowest serial number on the SPM's sending data flow to which a synchronization point confirmation is expected to be received. No confirmation is expected to be received when V(As) = V(Ms). V(Ar) is the lowest serial number on the SPM's receiving data flow for which a confirmation has not yet been sent. No confirmation will be sent by the SPM when V(Ar) = V(Mr).

5.9.4.2 V(Ms), V(Mr)

V(Ms) and V(Mr) are used by the SPM to maintain the next serial number to be used on the sending and receiving data flows.

V(Ms) is the serial number of the next synchronization point to be sent. V(Mr) is the serial number of the next synchronization point to be received.

5.9.4.3 V(Rs), V(Rr)

V(Rs) and V(Rr) are used by the SPM to maintain the lowest serial numbers to which resynchronization restart is permitted.

V(Rs) is the lowest serial number on the SPM's sending data flow to which resynchronization restart is permitted.

V(Rr) is the lowest serial number on the SPM's receiving data flow to which resynchronization restart is permitted.

5.9.4.4 Vsc

Vsc is not used if the symmetric synchronize functional unit has been selected.

NOTE – The manipulation of V(As), V(Ar), V(Ms), V(Mr), V(Rs) and V(Rr) and the circumstances under which they are updated are specified in clause 7 and are summarized in Table A.5.

SECTION 2 – SESSION PROTOCOL SPECIFICATION

6 Use of the transport service

This clause defines the way that the transport service primitives are used by the SPM.

6.1 Assignment of a session connection to the transport connection

6.1.1 Purpose

Assignment of a session connection to a transport connection.

6.1.2 Transport service primitives

The procedure uses the following transport service primitives:

- T-CONNECT request;
- T-CONNECT indication;
- T-CONNECT response;
- T-CONNECT confirm;
- T-DISCONNECT request;
- T-DISCONNECT indication.

6.1.3 SPDUs used

No SPDUs are used during assignment to a transport connection.

6.1.4 Description

A session connection is assigned to an existing transport connection suitable for reuse, or a new transport connection is created for the purpose. This assignment is based on the quality of service (see ITU-T Rec. 215 | ISO/IEC 8326) requested by the SS-user in the S-CONNECT request.

If a transport connection is established with the transport expedited data option, the transport expedited flow is available for the duration of the transport connection. Use of transport expedited data is specified in 6.4.

The transport expedited flow is requested by the SPM when the T-CONNECT request is issued if:

a) the SS-user requested the expedited data functional unit; or

b) the SS-user requested an extended control QOS for the session connection.

Only the initiator of the transport connection is permitted to issue the CONNECT SPDU.

When a session connection is terminated, the underlying transport connection is also terminated, unless reuse of the transport connection has been agreed.

Use of the TS-user data parameter in T-CONNECT request, indication, response and confirm is reserved for future use. When a T-CONNECT request or a T-CONNECT response is issued, this parameter is empty. When a T-CONNECT indication or T-CONNECT confirm is received, this parameter is ignored.

6.2 Reuse of the transport connection

6.2.1 Purpose

To allow the transport connection to be retained for reuse by another session connection.

6.2.2 Transport service primitives

The procedure uses the following transport service primitives:

- T-DATA request;
- T-DATA indication.

6.2.3 SPDUs used

The following SPDUs are related to reuse of the transport connection:

- REFUSE SPDU (see 7.5);
- FINISH SPDU (see 7.6);
- DISCONNECT SPDU (see 7.7);
- ABORT SPDU (see 7.9);
- ABORT ACCEPT SPDU (see 7.10).

6.2.4 Description

When a session connection is refused, or has been successfully connected and subsequently disconnected, by abort or orderly release, the supporting transport connection may be either disconnected or reused.

The transport connection may be kept for reuse provided that the transport expedited flow is not available, and either:

a) the SPM which established the transport connection requests retention of the transport connection by parameter in an ABORT SPDU or a FINISH SPDU; or

b) the SPM which established the transport connection receives a REFUSE SPDU or an ABORT SPDU which indicates by parameter that the transport connection is to be retained.

To avoid contention for a retained transport connection, only the transport connection initiator may reuse the transport connection by sending a CONNECT SPDU to establish a new session connection.

6.3 Use of transport normal data

6.3.1 Purpose

To convey SPDUs in user data fields of transport service normal data primitives.

6.3.2 Transport service primitives

The procedure uses the following transport service primitives:

- T-DATA request;
- T-DATA indication.

6.3.3 SPDUs used

The following SPDUs are sent on the transport normal flow:

- CONNECT SPDU (see 7.1);
- OVERFLOW ACCEPT SPDU (see 7.2);
- CONNECT DATA OVERFLOW SPDU (see 7.3);
- ACCEPT SPDU (see 7.4);
- REFUSE SPDU (see 7.5);
- FINISH SPDU (see 7.6);
- DISCONNECT SPDU (see 7.7);
- NOT FINISHED SPDU (see 7.8);
- ABORT ACCEPT SPDU (see 7.10);

- DATA TRANSFER SPDU (see 7.11);
- TYPED DATA SPDU (see 7.13);
- CAPABILITY DATA SPDU (see 7.14);
- CAPABILITY DATA ACK SPDU (see 7.15);
- GIVE TOKENS SPDU (see 7.16);
- PLEASE TOKENS SPDU (see 7.17);
- GIVE TOKENS CONFIRM SPDU (see 7.18);
- GIVE TOKENS ACK SPDU (see 7.19);
- MINOR SYNC POINT SPDU (see 7.20);
- MINOR SYNC ACK SPDU (see 7.21);
- MAJOR SYNC POINT SPDU (see 7.22);
- MAJOR SYNC ACK SPDU (see 7.23);
- RESYNCHRONIZE SPDU (see 7.24);
- RESYNCHRONIZE ACK SPDU (see 7.25);
- EXCEPTION REPORT SPDU (see 7.27);
- EXCEPTION DATA SPDU (see 7.28);
- ACTIVITY START SPDU (see 7.29);
- ACTIVITY RESUME SPDU (see 7.30);
- ACTIVITY INTERRUPT SPDU (see 7.31);
- ACTIVITY INTERRUPT ACK SPDU (see 7.32);
- ACTIVITY DISCARD SPDU (see 7.33);
- ACTIVITY DISCARD ACK SPDU (see 7.34);
- ACTIVITY END SPDU (see 7.35);
- ACTIVITY END ACK SPDU (see 7.36).

If the SS-user data exceeds 9 octets or if the transport expedited flow is not available, the following additional SPDU is sent on the transport normal flow:

- ABORT SPDU (see 7.9).

6.3.4 Transfer of SPDUs

The SPDUs listed in 6.3.3 are transferred using the transport normal data transfer service.

6.3.5 Segmenting

Segmenting of SSDUs takes place under the following circumstances:

a) when a maximum TSDU size has been selected, in which case a data SSDU or a typed data SSDU may be mapped onto more than one SPDU;

b) when Protocol Version 2 is proposed or selected and either:

   1) the SPDU size would exceed the maximum TSDU size; or

   2) the SPDU size would exceed 65539 octets for an SPDU to be sent on the transport normal flow or 16 octets for an SPDU to be sent on the transport expedited flow,

in which case SSDUs other than data SSDUs, typed data SSDUs and expedited data SSDUs are mapped onto more than one SPDU.

In all other cases, each SSDU is mapped one-to-one onto an SPDU.

NOTE – Implementors should note that when segmenting is selected:

a) the control information of each SPDU indicates whether or not it contains the first or last segment of the SSDU;

b) the size of the segments of the SSDU is constrained by the maximum TSDU size selected for that direction of transfer.

6.3.6 Maximum TSDU size

When a maximum TSDU size has been selected, the SPDU size may not exceed the maximum TSDU size selected for that direction of transfer and a sequence of concatenated SPDUs may not exceed the maximum TSDU size selected for that direction of transfer.

6.3.7 Concatenation

Each SPDU is defined in Table 6 as belonging to one of the following categories:

a) Category 0 SPDUs which may be mapped one-to-one onto a TSDU or may be concatenated with one or more category 2 SPDUs;

b) Category 1 SPDUs which are always mapped one-to-one onto a TSDU;

c) Category 2 SPDUs which are never mapped one-to-one onto a TSDU.

Basic concatenations of a category 0 SPDU with a single category 2 SPDU, defined as valid and in the order indicated in Table 7, may always be mapped onto a single TSDU.

If the receiving SPM has indicated that it can accept extended concatenation, the sending SPM may map a category 0 SPDU with one or more category 2 SPDUs (as specified in Table 8) onto a single TSDU.

In the case where this concatenated sequence does not fit into a single TSDU, extended concatenation cannot be applied.

The valid mappings of SPDUs onto TSDUs are illustrated in Figure 2.

Any other concatenation of SPDUs is defined as invalid.

Table 6 – Category 0, 1 and 2 SPDUs

Category 0 SPDUs	Category 1 SPDUs	Category 2 SPDUs
GIVE TOKENS SPDU PLEASE TOKENS SPDU	CONNECT SPDU OVERFLOW ACCEPT SPDU CONNECT DATA OVERFLOW SPDU ACCEPT SPDU REFUSE SPDU FINISH SPDU DISCONNECT SPDU NOT FINISHED SPDU ABORT SPDU ABORT ACCEPT SPDU GIVE TOKENS CONFIRM SPDU GIVE TOKENS ACK SPDU PREPARE SPDU TYPED DATA SPDU	DATA TRANSFER SPDU MINOR SYNC POINT SPDU MINOR SYNC ACK SPDU MAJOR SYNC POINT SPDU MAJOR SYNC ACK SPDU RESYNCHRONIZE SPDU RESYNCHRONIZE ACK SPDU ACTIVITY START SPDU ACTIVITY RESUME SPDU ACTIVITY DISCARD SPDU ACTIVITY DISCARD ACK SPDU ACTIVITY INTERRUPT SPDU ACTIVITY INTERRUPT ACK SPDU ACTIVITY END SPDU ACTIVITY END ACK SPDU CAPABILITY DATA SPDU CAPABILITY DATA ACK SPDU EXCEPTION REPORT SPDU EXCEPTION DATA SPDU

Table 7 – Valid basic concatenation of SPDUs

First SPDU	Second SPDU
GIVE TOKENS SPDU	DATA TRANSFER SPDU
GIVE TOKENS SPDU PLEASE TOKENS SPDU	MINOR SYNC POINT SPDU MINOR SYNC ACK SPDU
GIVE TOKENS SPDU PLEASE TOKENS SPDU	MAJOR SYNC POINT SPDU MAJOR SYNC ACK SPDU
GIVE TOKENS SPDU[a)] PLEASE TOKENS SPDU	RESYNCHRONIZE SPDU RESYNCHRONIZE ACK SPDU
GIVE TOKENS SPDU PLEASE TOKENS SPDU	ACTIVITY START SPDU ACTIVITY RESUME SPDU
GIVE TOKENS SPDU[a)] PLEASE TOKENS SPDU	ACTIVITY DISCARD SPDU ACTIVITY DISCARD ACK SPDU
GIVE TOKENS SPDU[a)] PLEASE TOKENS SPDU	ACTIVITY INTERRUPT SPDU ACTIVITY INTERRUPT ACK SPDU
GIVE TOKENS SPDU PLEASE TOKENS SPDU	ACTIVITY END SPDU ACTIVITY END ACK SPDU
GIVE TOKENS SPDU[a)] PLEASE TOKENS SPDU	CAPABILITY DATA SPDU CAPABILITY DATA ACK SPDU
PLEASE TOKENS SPDU PLEASE TOKENS SPDU	EXCEPTION REPORT SPDU EXCEPTION DATA SPDU

a) Indicates that the Token Item parameter is not present in the GIVE TOKENS SPDU. In all other cases, the Token Item parameter may or may not be present.

In all cases, the Token Item parameter may only be present in the first SPDU if the second SPDU contains either a complete SSDU, or the last segment of a segmented SSDU.

Basic concatenation of a PLEASE TOKENS SPDU or a GIVE TOKENS SPDU with a second SPDU is only permitted when the user data parameter is not present in the PLEASE TOKENS SPDU or the GIVE TOKENS SPDU.

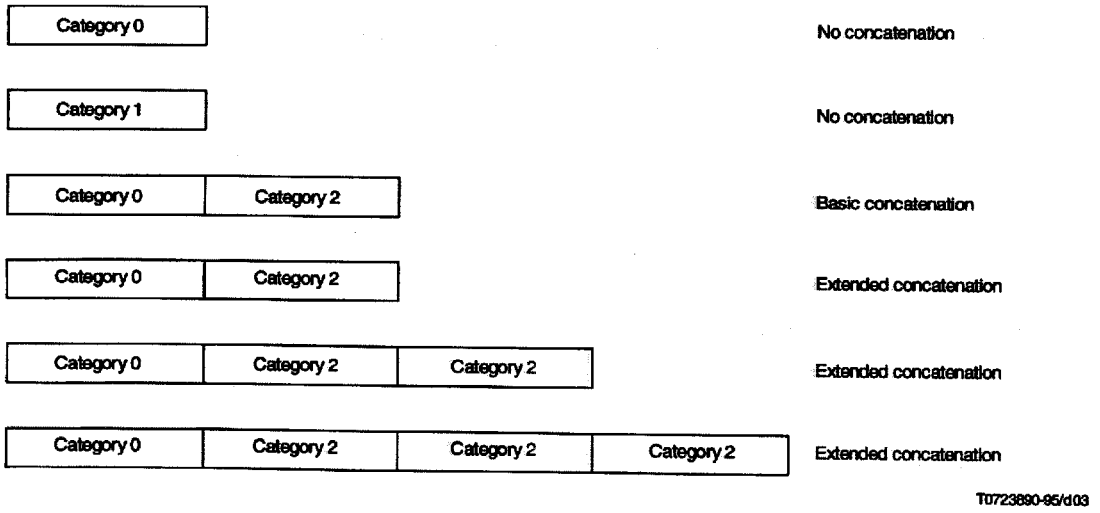

Figure 2 – Illustration of TSDU structures

6.3.7.1 Processing order of concatenated SPDUs

On receipt of SPDUs that have been concatenated using basic concatenation, the category 2 SPDUs are processed before the category 0 SPDU.

On receipt, SPDUs that have been concatenated using extended concatenation are processed in the following order:

a) ACTIVITY START SPDU; or

ACTIVITY RESUME SPDU;

b) DATA TRANSFER SPDU;

c) MINOR SYNC POINT SPDU; or

MINOR SYNC ACK SPDU; or

MAJOR SYNC POINT SPDU; or

MAJOR SYNC ACK SPDU; or

ACTIVITY END SPDU; or

ACTIVITY END ACK SPDU;

d) GIVE TOKENS SPDU; or

PLEASE TOKENS SPDU.

Table 8 – Valid extended concatenation of SPDUs

First SPDU	Second SPDU	Third SPDU	Fourth SPDU	Status
GIVE TOKENS SPDU GIVE TOKENS SPDU	MINOR SYNC ACK SPDU MAJOR SYNC ACK SPDU			
GIVE TOKENS SPDU	ACTIVITY END ACK SPDU			
GIVE TOKENS SPDU GIVE TOKENS SPDU GIVE TOKENS SPDU[a)] GIVE TOKENS SPDU[a)] GIVE TOKENS SPDU[a)] GIVE TOKENS SPDU[a)]	ACTIVITY START SPDU ACTIVITY RESUME SPDU ACTIVITY START SPDU ACTIVITY RESUME SPDU ACTIVITY START SPDU ACTIVITY RESUME SPDU	MINOR SYNC POINT SPDU MINOR SYNC POINT SPDU ACTIVITY END SPDU ACTIVITY END SPDU MAJOR SYNC POINT SPDU MAJOR SYNC POINT SPDU		
GIVE TOKENS SPDU GIVE TOKENS SPDU GIVE TOKENS SPDU GIVE TOKENS SPDU	MINOR SYNC POINT SPDU MINOR SYNC ACK SPDU MAJOR SYNC POINT SPDU MAJOR SYNC ACK SPDU	DATA TRANSFER SPDU DATA TRANSFER SPDU DATA TRANSFER SPDU DATA TRANSFER SPDU		CL CL CL CL
GIVE TOKENS SPDU GIVE TOKENS SPDU	ACTIVITY START SPDU ACTIVITY RESUME SPDU	DATA TRANSFER SPDU DATA TRANSFER SPDU		CF CF
GIVE TOKENS SPDU GIVE TOKENS SPDU	ACTIVITY END SPDU ACTIVITY END ACK SPDU	DATA TRANSFER SPDU DATA TRANSFER SPDU		CL CL
GIVE TOKENS SPDU GIVE TOKENS SPDU GIVE TOKENS SPDU[a)] GIVE TOKENS SPDU[a)] GIVE TOKENS SPDU[a)] GIVE TOKENS SPDU[a)]	ACTIVITY START SPDU ACTIVITY RESUME SPDU ACTIVITY START SPDU ACTIVITY RESUME SPDU ACTIVITY START SPDU ACTIVITY RESUME SPDU	MINOR SYNC POINT SPDU MINOR SYNC POINT SPDU ACTIVITY END SPDU ACTIVITY END SPDU MAJOR SYNC POINT SPDU MAJOR SYNC POINT SPDU	DATA TRANSFER SPDU DATA TRANSFER SPDU DATA TRANSFER SPDU DATA TRANSFER SPDU DATA TRANSFER SPDU DATA TRANSFER SPDU	C C C C C C

a) Indicates that the Token Item parameter is not present in the GIVE TOKENS SPDU.

CL The DATA TRANSFER SPDU contains a complete SSDU or the last segment of an SSDU.

CF The DATA TRANSFER SPDU contains a complete SSDU or the first segment of an SSDU. In the latter case, the Token Item parameter is not present in the GIVE TOKENS SPDU.

C The DATA TRANSFER SPDU contains a complete SSDU.

6.4 Use of transport expedited data

6.4.1 Purpose

To convey SPDUs on a separate transport flow.

6.4.2 Transport service primitives

The procedure uses the following transport service primitives:

- T-EXPEDITED-DATA request;
- T-EXPEDITED-DATA indication.

6.4.3 SPDUs used

The following SPDUs are sent on the transport expedited flow when it is available:

- ABORT SPDU (see 7.9);
- EXPEDITED DATA SPDU (see 7.12);
- PREPARE SPDU (see 7.26).

6.4.4 Description

The SPDUs listed in 6.4.3 are sent on the transport expedited flow if it is selected, and may be used to bypass any flow control restrictions or congestion on the transport normal flow. SPDUs sent on the transport expedited flow may be delivered to the accepting SS-user earlier than SSDUs submitted previously by the sending SS-user and sent on the transport normal flow, but no later than subsequently submitted SSDUs.

When the transport expedited flow is not available:

a) EXPEDITED DATA SPDUs are not sent;

b) ABORT SPDUs are sent on the transport normal flow;

c) PREPARE SPDUs are not sent.

6.5 Flow control

There is no peer flow control in the session layer. To prevent the SS-users from being overloaded with data, the receiving SPM may apply back pressure across the transport connection, using the transport flow control. The decision on when or how back pressure is applied is a local matter.

6.6 Transport disconnection

6.6.1 Purpose

To release a transport connection.

6.6.2 Transport service primitives

The procedure uses the following transport service primitives:

- T-DISCONNECT request;
- T-DISCONNECT indication.

6.6.3 SPDUs used

No SPDUs are used.

6.6.4 Description

After the session connection has been released or aborted and the transport connection is not to be reused, the transport connection is disconnected.

When a T-DISCONNECT indication is received, as a result of an error detected by the transport service provider, the SPM issues an S-P-ABORT indication to the local SS-user.

When issuing a T-DISCONNECT request, the SPM may optionally use the T-DISCONNECT user data field to indicate the reason for the transport disconnection to the remote SPM. The reason code consists of one octet with the following values:

a) 0 – Session protocol error for which an ABORT SPDU could not be sent.

b) 1 – Normal transport disconnection when the transport connection is not to be reused.

c) 2 – Normal transport disconnection when the transport connection was to be reused, but reuse is not possible for local reasons.

The use of the Disconnect Reason parameter in T-DISCONNECT indication is a local matter.

7 Elements of procedure related to SPDUs

This clause defines valid sequences of operation of the protocol.

A more precise definition of procedures is contained in Annex A which incorporates all the checks to determine the validity of a particular event at a particular point in time. In case of arbitration or dispute, Annex A takes precedence over this clause.

The elements of procedure specified in 7.4 to 7.8, 7.14 to 7.18, 7.20 to 7.25 and 7.28 to 7.36 do not consider the case where an SSDU is segmented. (The circumstances under which an SSDU may be segmented are specified in 6.3.5.) Additional elements of procedure for segmented SSDUs are specified in 7.37.

7.1 CONNECT SPDU

The CONNECT SPDU is transmitted by the initiator of the transport connection on a previously assigned transport connection in order to initiate a session connection.

7.1.1 Content of CONNECT SPDU

The CONNECT SPDU contains:

a) A Connection Identifier parameter group, which is supplied by the calling SS-user, to enable the SS-users to identify this specific session connection. This parameter group has no effect on the SPM. It contains:

   1) a Calling SS-user Reference parameter;

   2) a Common Reference parameter;

   3) An Additional Reference Information parameter.

b) A Connect/Accept Item parameter group containing:

   1) A Protocol Options parameter which enables the initiator to indicate its ability to receive extended concatenated SPDUs;

   2) A TSDU maximum size parameter which, if present and not zero, indicates the initiator's proposed values for the maximum TSDU sizes for each direction of transfer (see 5.8.6, 6.3.5). If this parameter is not present or is zero, the TSDU size is not limited.

   3) A Version Number parameter to identify all versions of this protocol which are supported and are suitable for this session connection.

      NOTE – Protocol Version 1 is not suitable if there are more than 512 octets of SS-user data in this SPDU.

   4) An Initial Serial Number parameter which is proposed by the calling SS-user in the case where the activity management functional unit is not proposed and any of the minor synchronize, symmetric synchronize, major synchronize, or resynchronize functional units are proposed. As an SS-user option, the Initial Serial Number parameter may be proposed even if the activity management

functional unit is proposed provided that any of the minor synchronize, symmetric synchronize, major synchronize or resynchronize functional units are also proposed. When the symmetric synchronize functional unit is proposed, the Initial Serial Number parameter is the proposed value of the First Initial Serial Number which is associated with the calling SS-user's sending direction of flow. This parameter is not present if the Upper Limit Serial Number parameter is present.

5) A Token Setting Item parameter supplied by the calling SS-user, which proposes the initial token positions for each token available on this connection, as derived from the functional units proposed in the Session User Requirements parameter (see Table 4). The initial token positions can be specified to be on the initiator's side or on the acceptor's side or the initiator can specify that the decision is to be made by the called SS-user.

6) A Second Initial Serial Number parameter which is proposed by the calling SS-user in the case where the activity management functional unit is not proposed and the symmetric synchronize functional unit is proposed. As an SS-user option, the Second Initial Serial Number parameter may be proposed even if the activity management functional unit is proposed provided that the symmetric synchronize functional unit is also proposed. The Second Initial Serial Number parameter proposes the serial number associated with the called SS-user's sending direction of flow. This parameter is not present if the Upper Limit Serial Number parameter is present.

7) An Upper Limit Serial Number parameter which can be proposed by the calling SS-user under the conditions where an Initial Serial Number can be proposed [see 7.1.1, b), 4)]. The Upper Limit Serial Number parameter is used to define the maximum value of the serial number. When the symmetric synchronize functional unit is proposed, the maximum value of the serial number applies to both directions of flow.

8) A Large Initial Serial Number parameter which can be proposed by the calling SS-user under the conditions where an Initial Serial Number can be proposed [see 7.1.1, b), 4)]. This parameter shall only be present if the Upper Limit Serial Number parameter is present.

9) A Large Second Initial Serial Number parameter which can be proposed by the calling SS-user under the conditions where a Second Initial Serial Number can be proposed [see 7.1.1, b), 4)]. This parameter shall only be present if the Upper Limit Serial Number parameter is present.

c) A Session User Requirements parameter containing a list of the functional units proposed by the calling SS-user. At least one of the half-duplex and the duplex functional units shall be proposed. The SPM is required to provide the associated protocol functions.

d) Calling Session Selector and Called Session Selector parameters corresponding to the calling SS-user and the called SS-user may be present and are derived from session addresses provided by the calling SS-user.

e) Either:

1) a User Data parameter which allows a limited (512 or less octets) amount of transparent user data to be passed from the calling SS-user to the called SS-user; or

2) an Extended User Data parameter which allows between 513 and 10 240 octets of transparent user data to be passed from the calling SS-user to the called SS-user. This parameter shall not be present if Protocol Version 1 is proposed.

Only one of these two parameters may be used on the CONNECT SPDU.

f) A Data Overflow parameter which shall be present if and only if there is more than 10 240 octets of SS-user data and which indicates to the responder that there is more SS-user data to follow. The first 10 240 octets of SS-user data are sent in the Extended User Data parameter. This parameter shall not be present if Protocol Version 1 is proposed.

7.1.2 Sending the CONNECT SPDU

An S-CONNECT request results in the assignment of a transport connection. When the transport connection is established, a CONNECT SPDU is sent on the transport normal flow. If the Data Overflow parameter was not present in the CONNECT SPDU, the SPM waits until it receives an ACCEPT SPDU or a REFUSE SPDU. If the Data Overflow parameter was present in the CONNECT SPDU, the SPM waits until it receives an OVERFLOW ACCEPT SPDU or a REFUSE SPDU.

7.1.3 Receiving the CONNECT SPDU

A valid incoming CONNECT SPDU which is acceptable to the receiving SPM and does not contain the Data Overflow parameter results in an S-CONNECT indication to an SS-user, according to the Called Session Selector parameter of the CONNECT SPDU. The SPM then waits for an S-CONNECT response from the called SS-user.

A valid incoming CONNECT SPDU which is acceptable to the receiving SPM, contains the Data Overflow parameter and provided that Protocol Version 2 is to be selected, results in the SPM sending an OVERFLOW ACCEPT SPDU; the SPM then waits until it receives a CONNECT DATA OVERFLOW SPDU. Otherwise the SPM sends a REFUSE SPDU (see 7.5).

7.2 OVERFLOW ACCEPT SPDU

The OVERFLOW ACCEPT SPDU is used by the SPM to request the remainder of the S-CONNECT request SS-user data. This SPDU shall not be used if Protocol Version 1 is proposed.

7.2.1 Content of the OVERFLOW ACCEPT SPDU

The OVERFLOW ACCEPT SPDU contains:

a) A TSDU Maximum Size parameter which, if present and not zero indicates that segmenting has been proposed by the responder (see 6.3.5). The responder proposes alternative values for the maximum TSDU sizes for each direction of transfer (see 5.8.6). These values may be larger than, smaller than or equal to the values supplied by the initiator in the CONNECT SPDU. The smaller value for each direction of transfer is used for the maximum TSDU size for that direction of transfer.

b) A Version Number parameter indicating that Protocol Version 2 is proposed (and selected).

7.2.2 Sending the OVERFLOW ACCEPT SPDU

A valid incoming CONNECT SPDU which contains the Data Overflow parameter results in the SPM sending an OVERFLOW ACCEPT SPDU. The SPM then waits until it receives a CONNECT DATA OVERFLOW SPDU.

7.2.3 Receiving the OVERFLOW ACCEPT SPDU

A valid incoming OVERFLOW ACCEPT SPDU results in the SPM sending one or more CONNECT DATA OVERFLOW SPDUs. When the last CONNECT DATA OVERFLOW SPDU has been sent, the SPM waits until it receives an ACCEPT SPDU or a REFUSE SPDU.

7.3 CONNECT DATA OVERFLOW SPDU

The CONNECT DATA OVERFLOW SPDU is used by the initiator to send subsequent segments of the User Data associated with an S-CONNECT request. This SPDU shall not be used if Protocol Version 1 has been selected.

7.3.1 Content of the CONNECT DATA OVERFLOW SPDU

The CONNECT DATA OVERFLOW SPDU contains:

a) an Enclosure Item parameter to indicate whether the SPDU is the middle or the end of the SSDU;

b) a User Data parameter which allows a maximum of 65 528 octets of transparent user data to be transferred.

7.3.2 Sending the CONNECT DATA OVERFLOW SPDU

A valid incoming OVERFLOW ACCEPT SPDU results in the SPM sending one or more CONNECT DATA OVERFLOW SPDUs. These SPDUs will be sent as an ordered sequence with the appropriate value for the Enclosure Item parameter until the complete SSDU has been transferred.

7.3.3 Receiving the CONNECT DATA OVERFLOW SPDU

A valid incoming CONNECT DATA OVERFLOW SPDU with Enclosure Item parameter indicating "end of SSDU" results in an S-CONNECT indication to pass the entire SSDU to the SS-user, according to the Called Session selector parameter of the CONNECT SPDU. The SPM then waits for an S-CONNECT response from the called SS-user.

If the Enclosure Item parameter in a valid incoming CONNECT DATA OVERFLOW SPDU indicates "not end of SSDU", the SPM waits for a subsequent valid CONNECT DATA OVERFLOW SPDU.

7.4 ACCEPT SPDU

An SPM receiving a CONNECT SPDU with the Data Overflow parameter absent may accept a proposal to establish a session connection by transferring an ACCEPT SPDU (after receiving an S-CONNECT response primitive) to the initiator, on the same transport connection.

An SPM which has previously issued an OVERFLOW ACCEPT SPDU in response to a CONNECT SPDU with the Data Overflow parameter present and which subsequently receives the sequence of CONNECT DATA OVERFLOW SPDUs which complete the segmented SSDU may accept a proposal to establish a session connection by transferring an ACCEPT SPDU (after receiving an S-CONNECT response primitive) to the initiator, on the same transport connection. An SPM receiving a CONNECT SPDU may accept a proposal to establish a session connection by transferring an ACCEPT SPDU to the initiator, on the same transport connection.

7.4.1 Content of ACCEPT SPDU

The ACCEPT SPDU contains:

a) A Connection Identifier parameter group, which is supplied by the called SS-user, to enable the SS-users to identify this specific session connection. This parameter group has no effect on the SPM. It contains:

   1) a Called SS-user Reference parameter;

   2) a Common Reference parameter;

   3) an Additional Reference Information parameter.

b) A Connect/Accept Item parameter group containing:

   1) A Protocol Options parameter which allows the responder to indicate its ability to receive extended concatenated SPDUs.

   2) A TSDU Maximum Size parameter which, if present and not zero, indicates the responder's proposed values for the maximum TSDU sizes for each direction of transfer (see 5.8.6 and 6.3.5). These values may be larger than, smaller than or equal to the values supplied by the initiator in the CONNECT SPDU. The smaller value is used for the maximum TSDU size for each direction of transfer. If an OVERFLOW ACCEPT SPDU has previously been sent on this session connection then:

      i) if the TSDU Maximum Size parameter was present in the OVERFLOW ACCEPT SPDU, then it shall also be present in the ACCEPT SPDU, with the same values as were given in the OVERFLOW ACCEPT SPDU;

      ii) if the TSDU Maximum Size parameter was not present in the OVERFLOW ACCEPT SPDU, then it shall not be present in the ACCEPT SPDU.

   3) A Version Number parameter to identify all versions of this protocol which are supported and are suitable for this session connection. The highest version number indicated by both initiator and responder is used. If an OVERFLOW ACCEPT SPDU has been previously sent on this session connection, then the Version Number parameter shall be present in the ACCEPT SPDU, with the same value as was given in the OVERFLOW ACCEPT SPDU.

      NOTE 1 – Protocol Version 1 is not suitable if there are more than 512 octets of User Data in this SPDU.

   4) An Initial Serial Number parameter which is present if the activity management functional unit is not selected and any of the minor synchronize, symmetric synchronize, major synchronize, or resynchronize functional units are selected regardless of whether or not the activity management functional unit is proposed. The called SS-user proposes the value, which is the value of the first serial number to be used. If the symmetric synchronize functional unit is selected, it is the value of the first serial number to be used on the calling SS-user's sending direction of flow. This parameter is not present if the Upper Limit Serial Number parameter is present.

   5) A Token Setting Item parameter supplied by the called SS-user, which indicates the initial token positions for each token available on this session connection, as derived from the selected functional units. A token is only available if any functional unit which requires that token has been selected for use on this session connection (see Table 4), regardless of the settings of the Token Setting Item parameter in the CONNECT SPDU [see 7.1.1, b), 4)]. If a token-controlled functional unit has been selected, then in the case where the calling SS-user has indicated that the initial assignment of the related token is the called SS-user's choice, this parameter contains a value chosen by the called SS-user. Otherwise, the values indicated by the calling SS-user in the CONNECT SPDU are selected and shall be returned.

   6) A Second Initial Serial Number parameter which is present if the activity management functional unit is not selected and the symmetric synchronize functional unit is selected regardless of whether or not the activity management functional unit is proposed. The called SS-user proposes the value, which is the value of the first serial number to be used on the called SS-user's sending direction of flow. This parameter is not present if the Upper Limit Serial Number parameter is present.

7) An Upper Limit Serial Number parameter which can be proposed by the called SS-user under the conditions where an Initial Serial Number can be proposed [see 7.4.1, b), 4)]. The Upper Limit Serial Number parameter is used to define the maximum value of the serial number. When the symmetric synchronize functional unit is proposed, the maximum value of the serial number applies to both directions of flow.

8) A Large Initial Serial Number parameter which can be proposed by the called SS-user under the conditions where an Initial Serial Number can be proposed [see 7.4.1, b), 4)]. This parameter shall only be present if the Upper Limit Serial Number parameter is present.

9) A Large Second Initial Serial Number parameter which can be proposed by the called SS-user under the conditions where a Second Initial Serial Number can be proposed [see 7.4.1, b), 4)]. This parameter shall only be present if the Upper Limit Serial Number parameter is present.

c) A Token Item parameter which allows the called SS-user to request tokens which have been assigned to the calling SS-user in the CONNECT SPDU.

NOTE 2 – This allows an implementation to convey a request for tokens in an ACCEPT SPDU, in the case when an S-CONNECT (accept) response is consecutively followed by an S-TOKEN-PLEASE request.

d) A Session User Requirements parameter which contains a list indicating the functional units proposed by the called SS-user and can be supported by the responder. The functional units selected for use on this session connection are the intersection of this set and the set proposed in the CONNECT SPDU (i.e. only those functional units indicated in both the CONNECT SPDU and the ACCEPT SPDU are selected). If both the half-duplex functional unit and the duplex functional unit were indicated in the CONNECT SPDU, then the ACCEPT SPDU shall propose which one is to be available. If only one of these functional units was indicated in the CONNECT SPDU, then the ACCEPT SPDU shall indicate that the same functional unit is to be used (or the connection attempt must be rejected). If both the minor synchronize and the symmetric synchronize functional units were indicated in the CONNECT SPDU, then the ACCEPT SPDU shall propose at most one of these two functional units.

e) An Enclosure Item parameter to indicate whether the SPDU is the beginning, middle or end of the SSDU (see also 7.37.1). This parameter shall not be present when Protocol Version 1 has been selected.

f) A Calling Session Selector parameter corresponding to the calling SS-user may be present, in which case it will have the same value as in the CONNECT SPDU. A Responding Session Selector parameter corresponding to the called SS-user may be present and is derived from the Responding Session Address provided by the called SS-user.

g) A User Data parameter which allows transparent user data to be passed from the called SS-user to the calling SS-user.

7.4.2 Sending the ACCEPT SPDU

An S-CONNECT (accept) response results in an ACCEPT SPDU. This SPDU is sent on the transport normal flow. After this successful connection, the SPM enters the data transfer phase and can receive any service request or SPDU that is allowed by the selected functional units and current token positions. If any of the minor synchronize, major synchronize or resynchronize functional units are selected but the symmetric synchronize functional unit is not selected and the activity management functional unit is not selected, the SPM sets V(A) and V(M) to the Initial Serial Number proposed by the called SS-user, which is the serial number to be used for the first synchronization point. V(R) is set to zero. Vsc is set false.

If the symmetric synchronize functional unit is selected but the activity management functional unit is not selected, the SPM sets V(Ar) and V(Mr) to the Initial Serial Number proposed by the called SS-user, which is the serial number to be used for the first received synchronization point. The SPM sets V(As) and V(Ms) to the Second Initial Serial Number proposed by the called SS-user, which is the serial number to be used for the first synchronization point to be sent. V(Rs) and V(Rr) are set to zero.

If the activity management functional unit has been selected, Vact is set false.

7.4.3 Receiving the ACCEPT SPDU

A valid incoming ACCEPT SPDU results in an S-CONNECT (accept) confirm. After this successful connection, the SPM enters the data transfer phase and can receive any service request or SPDU that is allowed by the selected functional units and current token positions. If any of the minor synchronize, major synchronize or resynchronize functional units are selected but the symmetric synchronize functional unit is not selected and the activity management functional unit is not selected, the SPM sets V(A) and V(M) to the Initial Serial Number contained in the ACCEPT SPDU, which is the serial number to be used for the first synchronization point. V(R) is set to zero. Vsc is set false.

If the symmetric synchronize functional unit is selected but the activity management functional unit is not selected, the SPM sets V(As) and V(Ms) to the Initial Serial Number contained in the ACCEPT SPDU, which is the serial number to be used for the first synchronization point to be sent. The SPM sets V(Ar) and V(Mr) to the Second Initial Serial Number contained in the ACCEPT SPDU, which is the serial number to be used for the first received synchronization point. V(Rs) and V(Rr) are set to zero.

If the activity management functional unit has been selected, Vact is set false.

If the ACCEPT SPDU contains the Token Item Parameter [see 7.4.1, c)], this shall be regarded by the SPM as if an ACCEPT SPDU had been received, followed by a PT SPDU. Therefore, when the Token Item parameter is present in the ACCEPT SPDU, an S-CONNECT (accept) confirm is followed by an S-TOKEN-PLEASE indication.

7.5 REFUSE SPDU

A REFUSE SPDU is used by the responder to reject an attempt to establish a session connection.

7.5.1 Content of REFUSE SPDU

The REFUSE SPDU contains:

a) A Connection Identifier parameter group, which is supplied by the called SS-user, to enable the SS-users to identify this specific session connection. This parameter group has no effect on the SPM. It contains:

   1) a Called SS-user Reference parameter;

   2) a Common Reference parameter;

   3) an Additional Reference Information parameter.

b) A Transport Disconnect parameter which indicates whether or not the transport connection is to be kept.

c) A Session User Requirements parameter which contains a list of the functional units supported by the sending SPM, and required by the called SS-user.

d) A Version Number parameter to identify which versions of this protocol have been implemented by the sending SPM.

e) An Enclosure Item parameter to indicate whether the SPDU is the beginning, middle or end of the SSDU (see also 7.37.1). This parameter shall not be present when Protocol Version 1 has been selected.

f) A Reason Code parameter giving the reason for refusal of the attempt to establish a session connection, together with transparent user data.

7.5.2 Sending the REFUSE SPDU

An S-CONNECT (reject) response results in a REFUSE SPDU. This SPDU is sent on the transport normal flow. No session connection is established. If the Transport Disconnect parameter indicates that the transport connection can be reused, the SPM waits for a CONNECT SPDU. Otherwise, the SPM starts the timer, TIM, and waits for a T-DISCONNECT indication. If the timer expires before receipt of a T-DISCONNECT indication, the SPM requests transport disconnection with a T-DISCONNECT request. The timer is cancelled on receipt of a T-DISCONNECT indication.

NOTE – The value of TIM is a local implementation dependent matter, related to Quality of Service.

7.5.3 Receiving the REFUSE SPDU

A valid incoming REFUSE SPDU results in an S-CONNECT (reject) confirm with the Responding Session Address parameter set to the value of the Called Session Address provided in the S-CONNECT request. No session connection is established. If the Transport Disconnect parameter indicates that retention of the transport connection has been requested by the called SPM, and this is acceptable to the calling SPM, the SPM waits for an S-CONNECT request. Otherwise, the SPM releases the transport connection, by making a T-DISCONNECT request.

7.6 FINISH SPDU

Orderly release is initiated by transfer of a FINISH SPDU, which may be transferred during the data transfer phase. It requests as a response either:

a) a DISCONNECT SPDU to complete the release of the session connection; or

b) a NOT FINISHED SPDU to refuse the release of the session connection if the release token is available.

The FINISH SPDU is transferred in sequence with any normal data being transferred. The right to issue a FINISH SPDU is restricted to the owner of all available tokens.

7.6.1 Content of FINISH SPDU

The FINISH SPDU contains:

a) A Transport Disconnect parameter which indicates whether or not the transport connection is to be kept, subject to the restrictions specified in 6.2.4.

b) An Enclosure Item parameter to indicate whether the SPDU is the beginning, middle or end of the SSDU (see also 7.37.1). This parameter shall not be present when Protocol Version 1 has been selected.

c) A User Data parameter which allows transparent user data to be transferred.

7.6.2 Sending the FINISH SPDU

An S-RELEASE request results in a FINISH SPDU. This SPDU is sent on the transport normal flow. After transferring a FINISH SPDU, the SPM may not send any further SPDUs (except ABORT SPDU or, in the case of collision of FINISH SPDUs, a DISCONNECT SPDU) unless a NOT FINISHED SPDU or a RESYNCHRONIZE SPDU is received, after which the data transfer phase may be resumed. Receipt of a DISCONNECT SPDU signals completion of orderly session release.

7.6.3 Receiving the FINISH SPDU

A valid incoming FINISH SPDU results in an S-RELEASE indication. The user data is passed to the SS-user. The SPM waits for an S-RELEASE response.

7.7 DISCONNECT SPDU

After receipt of a FINISH SPDU, a DISCONNECT SPDU may be transferred. Receipt of a DISCONNECT SPDU after transferring a FINISH SPDU signals the orderly release of the session connection. The DISCONNECT SPDU is transferred in sequence with any normal data being transferred.

7.7.1 Content of DISCONNECT SPDU

The DISCONNECT SPDU contains:

a) An Enclosure Item parameter to indicate whether the SPDU is the beginning, middle or end of the SSDU (see also 7.37.1). This parameter shall not be present when Protocol Version 1 has been selected.

b) A User Data parameter which allows transparent user data to be transferred.

7.7.2 Sending the DISCONNECT SPDU

An S-RELEASE (accept) response results in a DISCONNECT SPDU. This SPDU is sent on the transport normal flow. The session connection ceases to exist.

If the FINISH SPDU indicated that the transport connection is to be kept for reuse, and this is acceptable, the SPM waits for a CONNECT SPDU. Otherwise, the SPM starts the timer, TIM, and waits for a T-DISCONNECT indication. If the timer expires before receipt of a T-DISCONNECT indication, the SPM requests transport disconnection with a T-DISCONNECT request. The timer is cancelled on receipt of a T-DISCONNECT indication.

NOTE – The value of TIM is a local implementation dependent matter, related to Quality of Service.

7.7.3 Receiving the DISCONNECT SPDU

A valid incoming DISCONNECT SPDU results in an S-RELEASE (accept) confirm. The session connection ceases to exist.

If the transport connection is to be kept for reuse (see 6.2.4), the SPM waits for a suitable S-CONNECT request. Otherwise, a T-DISCONNECT request is issued.

NOTES

1 In the case of collision of FINISH SPDU and ABORT SPDU (see 7.9), the ABORT SPDU takes preference and thus the indication in the FINISH SPDU to keep or release the transport connection is ignored.

2 In the case of collision of FINISH SPDUs (data token and release token not available), the transport connection cannot be reused. The SPM receiving the DISCONNECT SPDU issues a T-DISCONNECT request.

7.8 NOT FINISHED SPDU

After receipt of a FINISH SPDU, a NOT FINISHED SPDU may be transferred subject to the token restrictions specified in Table 5. No confirmation is sought.

7.8.1 Content of NOT FINISHED SPDU

The NOT FINISHED SPDU contains:

a) An Enclosure Item parameter to indicate whether the SPDU is the beginning, middle or end of the SSDU (see also 7.37.1). This parameter shall not be present when Protocol Version 1 has been selected.

b) A User Data parameter which allows transparent user data to be transferred.

7.8.2 Sending the NOT FINISHED SPDU

An S-RELEASE (reject) response results in a NOT FINISHED SPDU. This SPDU is sent on the transport normal flow. The SPM remains in the data transfer phase and can receive any service request or SPDU that is allowed by the available functional units and current token positions.

7.8.3 Receiving the NOT FINISHED SPDU

A valid incoming NOT FINISHED SPDU results in an S-RELEASE (reject) confirm. The SPM remains in the data transfer phase and can receive any service request or SPDU that is allowed by the available functional units and current token positions.

7.9 ABORT SPDU

The ABORT SPDU is used to reject a session connection establishment attempt, or to cause abnormal release of a session connection at any time. This SPDU is also used by an SPM to release the session connection when a protocol error is detected. The ABORT SPDU may or may not request that the transport connection be released by the receiving SPM. Use of the ABORT SPDU may result in loss of data.

7.9.1 Content of ABORT SPDU

7.9.1.1 If there is no SSDU or segmenting of the SSDU is not required (see 6.3.5), the ABORT SPDU contains:

a) a Transport Disconnect parameter which indicates whether or not the transport connection is to be kept;

b) a Reflect Parameter Values parameter which, if present, allows implementation defined information to be transferred;

c) a User Data parameter which, if present, allows transparent user data to be transferred.

7.9.1.2 If the SSDU is to be segmented, the first ABORT SPDU contains:

a) a Transport Disconnect parameter which indicates whether or not the transport connection is to be kept;

b) an Enclosure Item parameter which indicates that this SPDU is the beginning of the SSDU and not the end of the SSDU;

c) a User Data parameter which allows transparent user data to be transferred.

The second and any subsequent ABORT SPDUs in the sequence of ABORT SPDUs transmitting the SSDU contain:

d) an Enclosure Item parameter to indicate whether the SPDU is the middle or end of the SSDU;

e) a User Data parameter which allows transparent user data to be transferred.

7.9.2 Sending the ABORT SPDU

An S-U-ABORT request or the detection of a protocol error in any state of the SPM results in either a single ABORT SPDU or, if the SSDU provided in the S-U-ABORT request is to be segmented (see 6.3.5), a sequence of ABORT SPDUs, which shall not be interrupted.

If the SS-user data does not exceed 9 octets, the ABORT SPDU is sent on the transport expedited flow, if it is available to this session connection. If the transport expedited flow is not available to this session connection, this SPDU is sent on the transport normal flow.

If the SS-user data exceeds 9 octets, the SPDU or sequence of SPDUs are sent on the transport normal flow. If the transport expedited flow is available to this session connection, a PREPARE (ABORT) SPDU is sent on the transport expedited flow simultaneously, or earlier than, the first, or only, ABORT SPDU.

The SPM starts the timer, TIM, and waits for an ABORT ACCEPT SPDU or a T-DISCONNECT indication. Any other SPDUs are discarded. If the timer expires before receipt of an ABORT ACCEPT SPDU or a T-DISCONNECT indication, the SPM shall request transport disconnection with a T-DISCONNECT request. On receipt of a T-DISCONNECT indication, the timer is cancelled.

NOTE – The value of TIM is a local implementation dependent matter, related to Quality of Service.

7.9.3 Receiving the ABORT SPDU

A valid incoming ABORT SPDU, without an Enclosure Item parameter, or an Enclosure item parameter indicating "end of SSDU" results in an S-U-ABORT indication or an S-P-ABORT indication, depending on whether the abort is user generated or provider generated. The session connection ceases to exist. If the Transport Disconnect parameter in the received ABORT SPDU indicates that the transport connection is to be kept for reuse and this is acceptable to the receiving SPM, an ABORT ACCEPT SPDU is sent. If the Transport Disconnect parameter in the received ABORT SPDU indicates that the transport connection is not to be kept for reuse or reuse of the transport connection is not acceptable to the receiving SPM, the receiving SPM either:

a) releases the transport connection; or

b) if the transport expedited flow is not available to this session connection, sends an ABORT ACCEPT SPDU (see 7.10).

Receiving an ABORT SPDU sent in response to a CONNECT SPDU results in:

a) a T-DISCONNECT request, unless retention of the transport connection has been requested in the ABORT SPDU, in which case the ABORT SPDU is acknowledged with an ABORT ACCEPT SPDU (see 7.10); and

b) an S-P-ABORT indication or an S-U-ABORT indication to the SS-user.

7.10 ABORT ACCEPT SPDU

The ABORT ACCEPT SPDU is used to return a confirmation to the ABORT SPDU when the transport expedited flow is not available to this session connection.

7.10.1 Content of ABORT ACCEPT SPDU

The ABORT ACCEPT SPDU contains no parameters.

7.10.2 Sending the ABORT ACCEPT SPDU

A valid incoming ABORT SPDU results in sending an ABORT ACCEPT SPDU, when the transport connection can be reused, i.e. when:

a) the transport expedited service is not available to this session connection; and

b) retention of the transport connection has been requested in the ABORT SPDU and it is acceptable to reuse the transport connection.

The SPM, as a local implementation decision, may send an ABORT ACCEPT SPDU in response to an ABORT SPDU, even if the transport connection is not to be kept, provided that the transport expedited flow is not available to this session connection.

This SPDU is sent on the transport normal flow. The session connection ceases to exist.

7.10.3 Receiving the ABORT ACCEPT SPDU

A valid incoming ABORT ACCEPT SPDU results in resetting the timer, TIM, and:

a) releasing the transport connection, if release of the transport connection was requested in the previously sent ABORT SPDU;

b) if retention of the transport connection was requested, the transport connection is now available for reuse by a new session connection, if this SPM was the initiator of the transport connection (see 6.1).

The session connection ceases to exist.

7.11 DATA TRANSFER SPDU

Normal data is transferred by use of the DATA TRANSFER SPDU. If the extended concatenation option was selected during connection establishment, certain concatenations of the DATA TRANSFER SPDU with other SPDUs is allowed (see 6.3.7).

The right to issue a DATA TRANSFER SPDU is subject to the token restrictions specified in Table 5.

7.11.1 Content of DATA TRANSFER SPDU

The DATA TRANSFER SPDU contains:

a) An Enclosure Item parameter to indicate the beginning and end of SSDU when segmenting has been selected. When segmenting has been selected, the Enclosure Item parameter is always present and indicates whether the SPDU is the beginning, middle or end of the SSDU. When segmenting has not been selected, the Enclosure Item parameter is not present.

b) A User Information Field to transfer transparent user data whose maximum size is unlimited when segmenting has not been selected and whose maximum size is limited by the maximum TSDU size when segmenting has been selected.

7.11.2 Sending the DATA TRANSFER SPDU

An S-DATA request results in a DATA TRANSFER SPDU unless segmenting has been selected, in which case an ordered sequence of DATA TRANSFER SPDUs will be sent with the appropriate value for the Enclosure Item parameter until the complete SSDU has been transferred.

The concatenation of any segment of an SSDU with any other SPDU will not result in a TSDU larger than the selected maximum TSDU size for that direction of transfer. However, there is no requirement that the resulting TSDU should be of the maximum size for that direction of transfer. All DATA TRANSFER SPDUs, except the last DATA TRANSFER SPDU in a sequence greater than one, must have user information. DATA TRANSFER SPDUs are sent on the transport normal flow.

Sending a segmented SSDU shall be interrupted when the SPM which is sending the segmented SSDU sends or receives one of:

- RESYNCHRONIZE SPDU;
- EXCEPTION REPORT SPDU;
- EXCEPTION DATA SPDU;
- ACTIVITY INTERRUPT SPDU;
- ACTIVITY DISCARD SPDU;
- ABORT SPDU;
- PREPARE (RESYNCHRONIZE) SPDU;
- PREPARE (ABORT) SPDU,

or receives a T-DISCONNECT indication. This will have a destructive effect on the entire SSDU. The SPM is not required to send the remainder of the ordered sequence of SPDUs which comprise the segmented SSDU (but may do so if it wishes).

7.11.3 Receiving the DATA TRANSFER SPDU

A valid incoming DATA TRANSFER SPDU results in an S-DATA indication unless segmenting has been selected. In this case, a valid incoming DATA TRANSFER SPDU, which indicates end of SSDU, results in an S-DATA indication to pass the entire SSDU to the SS-user.

Receiving a segmented SSDU shall be interrupted when the SPM which is receiving the segmented SSDU sends or receives one of:

- RESYNCHRONIZE SPDU;
- EXCEPTION REPORT SPDU;

- EXCEPTION DATA SPDU;
- ACTIVITY INTERRUPT SPDU;
- ACTIVITY DISCARD SPDU;
- ABORT SPDU;
- PREPARE (RESYNCHRONIZE) SPDU,

or receives a T-DISCONNECT indication. This will have a destructive effect on the entire SSDU (i.e. the SPDUs comprising part of the segmented SSDU which have already been received are discarded, and any SPDUs comprising part of the segmented SSDU which are received subsequently are discarded).

It is also valid to receive EXPEDITED SPDUs and PREPARE (MAJOR SYNC ACK) SPDUs and this has no effect on the segmented SSDU being received.

The receipt of any other SPDUs is a protocol error.

7.12 EXPEDITED SPDU

The EXPEDITED SPDU is used to transfer expedited SSDUs.

The right to send expedited data is not associated with any tokens. When this functional unit is selected, both SS-users may send expedited data. An expedited SSDU may be delivered to the receiving SS-user prior to other SSDUs previously transferred on the transport normal flow; it may not be delivered to the receiving SS-user later than any SSDUs transferred after it.

Expedited SSDUs are delivered to the receiving SS-user in the same sequence in which they were issued by the sending SS-user.

7.12.1 Content of EXPEDITED SPDU

The EXPEDITED SPDU contains a User Information Field which allows a limited amount of transparent user data to be transferred.

7.12.2 Sending the EXPEDITED SPDU

An S-EXPEDITED-DATA request results in an EXPEDITED SPDU being sent. This SPDU is sent on the transport expedited flow.

7.12.3 Receiving the EXPEDITED SPDU

A valid incoming EXPEDITED SPDU results in an S-EXPEDITED-DATA indication.

7.13 TYPED DATA SPDU

The TYPED DATA SPDU enables the SS-users to transmit transparent user data, irrespective of the availability or assignment of the data token. In all other respects, the same constraints apply as for normal data (see 7.11). The same rules for segmenting also apply.

7.13.1 Content of TYPED DATA SPDU

The TYPED DATA SPDU contains:

a) An Enclosure Item parameter to indicate the beginning and end of SSDU when segmenting has been selected. When segmenting has been selected, the Enclosure Item parameter is always present and indicates whether the SPDU is the beginning, middle or end of the SSDU. When segmenting has not been selected, the Enclosure Item parameter is not present.

b) A User Information Field to transfer transparent user data whose maximum size is unlimited when segmenting has not been selected and whose maximum size is limited by the maximum TSDU size when segmenting has been selected.

7.13.2 Sending the TYPED DATA SPDU

An S-TYPED-DATA request results in the transfer of a TYPED DATA SPDU unless segmenting has been selected, in which case an ordered sequence of TYPED DATA SPDUs will be sent with the appropriate value for the Enclosure Item parameter until the complete SSDU has been transferred. Each SPDU is mapped onto one TSDU and will not be larger than the selected maximum TSDU size for that direction of transfer. However, there is no requirement that the resulting TSDU should be of the maximum size for that direction of transfer. All TYPED DATA SPDUs, except the last TYPED DATA SPDU in a sequence greater than one, must have user information. TYPED DATA SPDUs are sent on the transport normal flow. When segmenting has been selected the rules governing the sending or receipt of SPDUs other than TYPED DATA SPDUs, while sending a segmented TYPED DATA SSDU are the same as for the DATA TRANSFER SPDU (see 7.11.2).

7.13.3 Receiving the TYPED DATA SPDU

A valid incoming TYPED DATA SPDU results in an S-TYPED-DATA indication, unless segmenting has been selected. In this case, a valid incoming TYPED DATA SPDU which indicates end of SSDU results in an S-TYPED-DATA indication to pass the entire SSDU to the SS-user. The current state of the SPM is not changed.

When segmenting has been selected the rules governing the sending or receipt of SPDUs other than TYPED DATA SPDUs, while receiving a segmented TYPED DATA SSDU are the same as for the DATA TRANSFER SPDU (see 7.11.3).

7.14 CAPABILITY DATA SPDU

The CAPABILITY DATA SPDU is used to transfer transparent user data outside activities (i.e. when the activity management functional unit has been selected and Vact is false). The right to send this SPDU is restricted to the side having the right to start the next activity (i.e. the activity management functional unit has been selected and Vact is false and subject to the token restrictions specified in Table 5).

7.14.1 Content of CAPABILITY DATA SPDU

The CAPABILITY DATA SPDU contains:

a) An Enclosure Item parameter to indicate whether the SPDU is the beginning, middle or end of the SSDU (see also 7.37.1). This parameter shall not be present when Protocol Version 1 has been selected.

b) A User Data parameter which allows transparent user data to be transferred.

7.14.2 Sending the CAPABILITY DATA SPDU

An S-CAPABILITY-DATA request results in a CAPABILITY DATA SPDU being sent. This SPDU is sent on the transport normal flow. The SS-user is not permitted to issue a further S-CAPABILITY-DATA request until this CAPABILITY DATA SPDU is acknowledged.

7.14.3 Receiving the CAPABILITY DATA SPDU

A valid incoming CAPABILITY DATA SPDU results in an S-CAPABILITY-DATA indication to the SS-user.

7.15 CAPABILITY DATA ACK SPDU

The CAPABILITY DATA ACK SPDU is used to complete the capability data exchange.

7.15.1 Content of CAPABILITY DATA ACK SPDU

The CAPABILITY DATA ACK SPDU contains:

a) An Enclosure Item parameter to indicate whether the SPDU is the beginning, middle or end of the SSDU (see also 7.37.1). This parameter shall not be present when Protocol Version 1 has been selected.

b) A User Data parameter which allows transparent user data to be transferred.

7.15.2 Sending the CAPABILITY DATA ACK SPDU

The SS-user generates an S-CAPABILITY-DATA response which results in a CAPABILITY DATA ACK SPDU. This SPDU is sent on the transport normal flow.

7.15.3 Receiving the CAPABILITY DATA ACK SPDU

A valid incoming CAPABILITY DATA ACK SPDU results in an S-CAPABILITY-DATA confirm. This allows the SS-user to issue a further S-CAPABILITY-DATA request.

7.16 GIVE TOKENS SPDU

The GIVE TOKENS SPDU is used:

a) to introduce a concatenated sequence of SPDUs; and/or

b) to cause assignment of currently owned tokens to be changed.

If the GIVE TOKENS SPDU does not contain any parameter fields, it is used to indicate concatenation without assignment of tokens and, in this case, the sending and receiving procedures do not apply.

7.16.1 Content of GIVE TOKENS SPDU

The GIVE TOKENS SPDU contains:

a) A Token Item parameter which indicates which tokens are being transferred from the sending SS-user to the receiving SS-user;

b) An Enclosure Item parameter to indicate whether the SPDU is the beginning, middle or end of the SSDU (see also 7.37.1). This parameter shall not be present when Protocol Version 1 has been selected;

c) A User Data parameter which allows transparent user data to be transferred. This parameter shall not be present if either:

1) Protocol Version 1 is selected; or

2) the GIVE TOKENS SPDU is being used to introduce a concatenated sequence of SPDUs.

7.16.2 Sending the GIVE TOKENS SPDU

An S-TOKEN-GIVE request results in a GIVE TOKENS SPDU. This SPDU is sent on the transport normal flow.

7.16.3 Receiving the GIVE TOKENS SPDU

A valid incoming GIVE TOKENS SPDU results in an S-TOKEN-GIVE indication.

7.17 PLEASE TOKENS SPDU

The PLEASE TOKENS SPDU is used:

a) to introduce a concatenated sequence of SPDUs; and/or

b) to request that the token assignments be changed to permit the requestor to be authorized to perform a function associated with the requested tokens.

If the PLEASE TOKENS SPDU does not contain any parameter fields, it is used to indicate concatenation without requesting tokens and, in this case, the sending and receiving procedures do not apply.

7.17.1 Content of PLEASE TOKENS SPDU

The PLEASE TOKENS SPDU contains:

a) A Token Item parameter which indicates which tokens are being requested by the sending SS-user.

b) An Enclosure Item parameter to indicate whether the SPDU is the beginning, middle or end of the SSDU (see also 7.37.1). This parameter shall not be present when Protocol Version 1 has been selected.

c) A User Data parameter which allows transparent user data to be transferred. This parameter shall not be present if the PLEASE TOKENS SPDU is being used to introduce a concatenated sequence of SPDUs.

7.17.2 Sending the PLEASE TOKENS SPDU

An S-TOKEN-PLEASE request results in a PLEASE TOKENS SPDU. This SPDU is sent on the transport normal flow.

7.17.3 Receiving the PLEASE TOKENS SPDU

A valid incoming PLEASE TOKENS SPDU results in an S-TOKEN-PLEASE indication. Receiving a PLEASE TOKENS SPDU for tokens which are not currently assigned to the accepting SS-user is not a protocol error.

7.18 GIVE TOKENS CONFIRM SPDU

The GIVE TOKENS CONFIRM SPDU is used as a result of an S-CONTROL-GIVE request to cause assignment of all of the currently assigned tokens to be changed when Vact is false. Receipt of the GIVE TOKENS CONFIRM SPDU by the receiving SPM is acknowledged by the GIVE TOKENS ACK SPDU.

7.18.1 Content of GIVE TOKENS CONFIRM SPDU

The GIVE TOKENS CONFIRM SPDU contains:

a) An Enclosure Item parameter to indicate whether the SPDU is the beginning, middle or end of the SSDU (see also 7.37.1). This parameter shall not be present when Protocol Version 1 has been selected.

b) A User Data parameter which allows transparent user data to be transferred. This parameter shall not be present if Protocol Version 1 is selected.

7.18.2 Sending the GIVE TOKENS CONFIRM SPDU

An S-CONTROL-GIVE request when Vact is false results in a GIVE TOKENS CONFIRM SPDU. The SPM then waits for a GIVE TOKENS ACK SPDU before permitting further SPDUs, associated with the available tokens, to be sent or received. SPDUs not associated with tokens (e.g. TYPED DATA SPDU) may be sent or received as normal. This SPDU is sent on the transport normal flow.

7.18.3 Receiving the GIVE TOKENS CONFIRM SPDU

A valid incoming GIVE TOKENS CONFIRM SPDU results in an S-CONTROL-GIVE indication, followed by a GIVE TOKENS ACK SPDU.

7.19 GIVE TOKENS ACK SPDU

The GIVE TOKENS ACK SPDU is used to acknowledge receipt of a GIVE TOKENS CONFIRM SPDU. The GIVE TOKENS ACK SPDU can only be sent when Vact is false.

7.19.1 Content of GIVE TOKENS ACK SPDU

The GIVE TOKENS ACK SPDU contains no parameters.

7.19.2 Sending the GIVE TOKENS ACK SPDU

A valid incoming GIVE TOKENS CONFIRM SPDU results in a GIVE TOKENS ACK SPDU (see also 7.18.3). The SPM may now transmit SPDUs associated with the token controlled functional units. This SPDU is sent on the transport normal flow.

7.19.3 Receiving the GIVE TOKENS ACK SPDU

After receiving a valid incoming GIVE TOKENS ACK SPDU, the SPM is now prepared to receive any SPDUs associated with the token controlled functional units.

7.20 MINOR SYNC POINT SPDU

The MINOR SYNC POINT SPDU is used to define a minor synchronization point. When the symmetric synchronize functional unit has been selected, the MINOR SYNC POINT SPDU is used to define a minor synchronization point on one direction of data flow. A confirmation may be returned by the receiver but is not required by the SPM (see 7.21). All acknowledgement rules are defined by the SS-users. In particular, whether confirmation is requested or not is transparent to the SPM. The right to issue a MINOR SYNC POINT SPDU is subject to the token restrictions specified in Table 5.

The serial numbers are managed by the SS-provider and given to the SS-user. The right to issue a MINOR SYNC POINT SPDU is not subject to any token restrictions with the symmetric synchronize functional unit.

7.20.1 Content of MINOR SYNC POINT SPDU

The MINOR SYNC POINT SPDU contains:

a) A Sync Type Item parameter which is used to indicate:

   1) if an explicit confirmation is required (see 7.21);

   2) if data separation is requested.

b) An Enclosure Item parameter to indicate whether the SPDU is the beginning, middle or end of the SSDU (see also 7.37.1). This parameter shall not be present when Protocol Version 1 has been selected.

c) A Serial Number parameter which indicates the serial number of this minor synchronization point. This parameter is set by the sending SPM to the current value of V(Ms) when symmetric synchronize functional unit has been selected, and to the current value of V(M) when the symmetric synchronize functional unit has not been selected.

d) A User Data parameter which allows transparent user data to be transferred.

7.20.2 Sending the MINOR SYNC POINT SPDU

An S-SYNC-MINOR request results in a MINOR SYNC POINT SPDU. This SPDU is sent on the transport normal flow. When the symmetric synchronize functional unit has been selected, V(Ms) is incremented by one. When the symmetric synchronize functional unit is not selected, variables are processed as follows. If Vsc is true, V(A) is set equal to V(M) and Vsc is set false. If the Sync Type parameter has the value "data separation", V(Ado) is set to V(M). V(M) is incremented by one.

7.20.3 Receiving the MINOR SYNC POINT SPDU

A valid incoming MINOR SYNC POINT SPDU results in an S-SYNC-MINOR indication. When the symmetric synchronize functional unit has been selected, V(Mr) is incremented by one. When the symmetric synchronize functional unit is not selected, variables are processed as follows. If Vsc is false, V(A) is set equal to V(M) and Vsc is set true. V(M) is incremented by one.

7.21 MINOR SYNC ACK SPDU

The MINOR SYNC ACK SPDU is used to return a confirmation to minor synchronization points. The SPM sends a MINOR SYNC ACK SPDU for each S-SYNC-MINOR response. When the symmetric synchronize functional unit has been selected, the local variable Vsc is not defined and its associated restrictions do not apply.

7.21.1 Content of MINOR SYNC ACK SPDU

The MINOR SYNC ACK SPDU contains:

a) An Enclosure Item parameter to indicate whether the SPDU is the beginning, middle or end of the SSDU (see also 7.37.1). This parameter shall not be present when Protocol Version 1 has been selected.

b) A Serial Number parameter, provided by the SS-user which indicates the serial number of the minor synchronization point which is being confirmed.

c) A User Data parameter which allows transparent user data to be transferred.

7.21.2 Sending the MINOR SYNC ACK SPDU

A valid S-SYNC-MINOR response results in sending a MINOR SYNC ACK SPDU. When the symmetric synchronize functional unit has been selected, the serial number is greater than or equal to V(Ar) and less than V(Mr); the SPM sets V(Ar) equal to the serial number plus one. When the symmetric synchronize function unit has not been selected, Vsc is true and the serial number is greater than or equal to V(A) and less than V(M); the SPM sets V(A) equal to the serial number plus one. The MINOR SYNC ACK SPDU is sent on the transport normal flow.

7.21.3 Receiving the MINOR SYNC ACK SPDU

A valid incoming MINOR SYNC ACK SPDU results in an S-SYNC-MINOR confirm. When the symmetric synchronize functional unit has been selected, the received serial number is greater than or equal to V(As) and less than V(Ms); the SPM sets V(As) equal to the received serial number plus one. When the symmetric synchronize functional unit has not been selected, Vsc is false and the received serial number is greater than or equal to V(A) and less than V(M); the SPM sets V(A) equal to the received serial number plus one.

7.22 MAJOR SYNC POINT SPDU

The MAJOR SYNC POINT SPDU is used to define a major synchronization point. A confirmation has to be received before more data can be sent on the normal and expedited flows. The right to issue a MAJOR SYNC POINT SPDU is subject to the token restrictions specified in Table 5.

If the symmetric synchronize functional unit has been selected, a major synchronization point is defined by two correlated serial numbers, one for each direction of data flow. One serial number is defined by the MAJOR SYNC POINT SPDU; this number is associated with the sending data flow of the requestor of the major synchronization point. The other serial number is defined in the MAJOR SYNC ACK SPDU. This number is associated with the receiving data flow of the requestor of the major synchronization point. Together these two serial numbers completely separate the data flow before and after the major synchronization point. Both serial numbers are managed by the SS-provider and given to the SS-user.

7.22.1 Content of MAJOR SYNC POINT SPDU

The MAJOR SYNC POINT SPDU contains:

a) A Sync Type Item parameter which is only present when indicating that this major synchronization point is not the end of the current activity.

b) An Enclosure Item parameter to indicate whether the SPDU is the beginning, middle or end of the SSDU (see also 7.37.1). This parameter shall not be present when Protocol Version 1 has been selected.

c) A Serial Number parameter which indicates the serial number of this major synchronization point. When the symmetric synchronize functional unit has been selected, this parameter indicates the First Serial Number, which is associated with the sending flow of the requestor of the major synchronization point, and is set by the sending SPM to the current value of V(Ms). It is set to the current value of V(M) when the symmetric synchronize functional unit has not been selected.

d) A User Data parameter which allows transparent user data to be transferred.

7.22.2 Sending the MAJOR SYNC POINT SPDU

An S-SYNC-MAJOR request results in a MAJOR SYNC POINT SPDU. This SPDU is sent on the transport normal flow. If the symmetric synchronize functional unit has been selected, V(Ms) is incremented by one. Otherwise, Vsc, V(A), and V(M) are updated as follows. If Vsc is true, V(A) is set equal to V(M) and Vsc is set false. V(M) is incremented by one.

If the activity management functional unit has been selected, Vnextact is set true. If the transport expedited flow is available to this session connection, the SPM waits for a PREPARE (MAJOR SYNC ACK) SPDU, followed by a MAJOR SYNC ACK SPDU. Otherwise, just a MAJOR SYNC ACK is expected. Any other SPDUs received prior to the MAJOR SYNC ACK SPDU will result in the appropriate service indications being given to the SS-user.

7.22.3 Receiving the MAJOR SYNC POINT SPDU

A valid incoming MAJOR SYNC POINT SPDU results in an S-SYNC-MAJOR indication. When the symmetric synchronize functional unit has been selected, the received serial number must be equal to V(Mr); and V(Mr) is incremented by one. If the symmetric synchronize functional unit has not been selected, the received serial number must be equal to V(M); and Vsc, V(A), and V(M) are updated as follows. If Vsc is false, V(A) is set equal to V(M). V(M) is incremented by one.

If the activity management functional unit has been selected, Vnextact is set true.

7.23 MAJOR SYNC ACK SPDU

The MAJOR SYNC ACK SPDU is used to return a confirmation to a major synchronization point. If the symmetric synchronize functional unit has been selected, this SPDU defines the serial number for the MAJOR SYNC POINT requestor's receiving flow.

7.23.1 Content of MAJOR SYNC ACK SPDU

The MAJOR SYNC ACK SPDU contains:

a) An Enclosure Item parameter to indicate whether the SPDU is the beginning, middle or end of the SSDU (see also 7.37.1). This parameter shall not be present when Protocol Version 1 has been selected.

b) A Serial Number parameter which indicates the serial number of the major synchronization point which is being confirmed [which is equal to V(M) minus one]. This parameter is only present with the symmetric synchronize functional unit has not been selected.

c) A Second Serial Number parameter which indicates the serial number of the major synchronization point for the receiving flow of the requestor of the major synchronization point, which is the direction of flow on which the MAJOR SYNC ACK SPDU travels. This number is set by the SPM to the current value of V(Ms). This parameter is only present when the symmetric synchronize functional unit has been selected.

d) A User Data parameter which allows transparent user data to be transferred.

7.23.2 Sending the MAJOR SYNC ACK SPDU

An S-SYNC-MAJOR response results in a MAJOR SYNC ACK SPDU. This SPDU is sent on the transport normal flow. If the transport expedited flow is available to this session connection, a PREPARE (MAJOR SYNC ACK) SPDU is sent simultaneously, or earlier, on the transport expedited flow. If the symmetric synchronize functional unit has been selected, V(Ms) is incremented by one; V(As) and V(Rs) are set to V(Ms); and V(Ar) and V(Rr) are set to V(Mr). If the symmetric synchronize functional unit has not been selected, V(A) and V(R) are set equal to V(M). If the activity management functional unit has been selected, Vact is set to Vnextact.

7.23.3 Receiving the MAJOR SYNC ACK SPDU

A valid incoming MAJOR SYNC ACK SPDU results in an S-SYNC-MAJOR confirm. When the symmetric synchronize functional unit has been selected, the received Second Serial Number must be equal to V(Mr). When the symmetric synchronize functional unit has not been selected, the received Serial Number must be equal to V(M) minus one.

If the transport expedited flow is available to this session connection, two successive SPDUs will be received:

a) PREPARE (MAJOR SYNC ACK) SPDU on the transport expedited flow, followed by

b) MAJOR SYNC ACK SPDU on the transport normal flow.

When the symmetric synchronize functional unit has been selected, V(Mr) is incremented; V(As) and V(Rs) are set equal to V(Ms); and V(Ar) and V(Rr) are set equal to V(Mr). When the symmetric synchronize function unit has not been selected, V(A) and V(R) are set equal to V(M). If the activity management functional unit has been selected, Vact is set to Vnextact.

7.24 RESYNCHRONIZE SPDU

The RESYNCHRONIZE SPDU is used to provide the SS-users with a selective means to resynchronize the exchange of data to a synchronization point and to reposition the tokens to an agreed side. Use of this procedure may result in loss of data.

This SPDU can also be used to "purge" the session connection, since that is a particular case of resynchronization. The following options are provided:

a) abandon;

b) set;

c) restart.

When the symmetric synchronize functional unit is selected, the SS-users may resynchronize the exchange of data to new serial numbers for both directions of flow, or they may resynchronize only one direction of flow. The SS-user specifies an option for each direction of flow on which resynchronization is requested. Except in the case where the option is abandon, the SS-user also specifies a serial number for each direction of flow on which resynchronization is requested.

Since the resynchronization protocol provides a repositioning of the tokens a particular use of it is the destructive way to get the tokens.

When used with activity management, the RESYNCHRONIZE SPDU can only be sent when Vact is true.

7.24.1 Content of RESYNCHRONIZE SPDU

The RESYNCHRONIZE SPDU contains:

a) An Enclosure Item parameter to indicate whether the SPDU is the beginning, middle or end of the SSDU (see also 7.37.1). This parameter shall not be present when Protocol Version 1 has been selected.

b) A Token Setting Item which indicates the requestor's proposed token positions for all available tokens.

c) A Resync Type parameter which indicates the resynchronize option (abandon, set or restart). If the symmetric synchronize functional unit has been selected, it indicates the First Resync Type, which is the resynchronize option for the requestor's sending flow.

d) A Serial Number parameter which indicates a serial number to which resynchronization is being requested. If the symmetric synchronize functional unit has been selected, it indicates the First Serial Number, which is the value of the serial number for the sending flow of the requestor of the Resynchronize. This parameter is supplied by the SS-user if the Resync Type is set or restart. If the Resync Type is abandon, the serial number is set to the value of V(Ms) of the sending SPM if the symmetric synchronize functional unit is selected, and to the value of V(M) if it is not selected.

e) A Second Resync Type parameter which indicates the resynchronize option for the requestor's receiving flow (abandon, set or restart). This parameter may only be present if the symmetric synchronize functional unit has been selected.

f) A Second Serial Number parameter which indicates the serial number for the requestor's receiving flow to which resynchronization is being requested. This serial number is supplied by the SS-user if the Second Resync Type is set or restart. If the Second Resync Type is abandon, the serial number is set to the value of V(Mr) of the sending SPM. This parameter may only be present if the symmetric synchronize functional unit has been selected.

g) A User Data parameter which allows transparent user data to be transferred.

7.24.2 Sending/receiving the RESYNCHRONIZE SPDU without symmetric synchronization

This subclause defines how the RESYNCHRONIZE SPDU is sent and received when the symmetric synchronize functional unit has not been selected.

7.24.2.1 Sending the RESYNCHRONIZE SPDU without symmetric synchronization

An S-RESYNCHRONIZE request [with serial number greater than or equal to V(R) and less than or equal to V(M) if the resynchronize option is restart] results in a RESYNCHRONIZE SPDU. This SPDU is sent on the transport normal flow. If the transport expedited flow is available to this session connection and the data separation functional unit has not been selected, a PREPARE (RESYNCHRONIZE) SPDU is sent simultaneously, or earlier, on the transport expedited flow. If the transport expedited flow is available to this session connection and the data separation functional unit has been selected, one of the following applies:

a) if a previously sent MINOR SYNCHRONIZATION POINT SPDU with the Sync Type Item having the value "data separation" has not been acknowledged, then a PREPARE (RESYNCHRONIZE) SPDU shall not be sent;

b) otherwise, a PREPARE (RESYNCHRONIZE) SPDU may be sent, as a local matter.

The SPM goes into a state where incoming SPDUs are discarded in the following way:

a) if the data separation functional unit is not selected or if there is no unacknowledged minor synchronization point with the Sync Type item having the value "data separation" (previously sent by the local SPM), all SPDUs are discarded except: PREPARE (RESYNCHRONIZE), RESYNCHRONIZE, PREPARE (RESYNCHRONIZE ACK), RESYNCHRONIZE ACK, ACTIVITY DISCARD, ACTIVITY INTERRUPT and ABORT SPDUs.

b) if the data separation functional unit is selected and there are unacknowledged minor synchronization points with the Sync Type item having the value "data separation" (previously sent by the local SPM), only the following SPDUs are discarded: PREPARE (MAJOR SYNC ACK) SPDUs, MAJOR SYNC ACK SPDUs (see the Note).

NOTE – The MAJOR SYNC ACK SPDU is discarded (and therefore does not result in an S-SYNC-MAJOR confirm) because the previously issued S-RESYNC request prevented the completion of the dialogue unit. It is not possible to deliver the S-SYNC-MAJOR confirm because this would cause V(R) to be updated and this may be a later value than that to which the resynchronize is occurring.

If a RESYNCHRONIZE, PREPARE (RESYNCHRONIZE), ACTIVITY INTERRUPT or ACTIVITY DISCARD SPDU is received when the SPM is in this state, a resynchronization contention situation has occurred and is dealt with as specified in 7.24.2.3.

7.24.2.2 Receiving the RESYNCHRONIZE SPDU without symmetric synchronization

Except when a resynchronization contention situation has occurred, a valid incoming RESYNCHRONIZE SPDU (with received serial number greater than or equal to V(R) if the resynchronize option is restart) results in an S-RESYNCHRONIZE indication. If the resynchronize option is abandon, this indication contains a serial number which is equal to V(M) or the received serial number, whichever is higher; V(M) is set to this value. If the transport expedited flow is available to this session connection, two successive SPDUs will be received:

a) PREPARE (RESYNCHRONIZE) SPDU on the transport expedited flow, followed by

b) RESYNCHRONIZE SPDU on the transport normal data flow.

When the PREPARE(RESYNCHRONIZE) SPDU is received, the SPM goes into a state where incoming SPDUs are discarded in the following way:

a) if the data separation functional unit is not selected or if there is no outstanding minor synchronization point with the Sync Type item having the value "data separation", all SPDUs are discarded, until the RESYNCHRONIZE SPDU is received, except for ABORT SPDUs.

b) if the data separation functional unit is selected and there are one or more outstanding minor synchronization points with the Sync Type item having the value "data separation" (previously sent by the SPM), only the following SPDUs are discarded, until the RESYNCHRONIZE SPDU is received: PREPARE (MAJOR SYNC ACK) SPDUs, MAJOR SYNC ACK SPDUs (see the Note).

NOTE – The MAJOR SYNC ACK SPDU is discarded (and therefore does not result in an S-SYNC-MAJOR confirm) because the previously issued S-RESYNC request prevented the completion of the dialogue unit. It is not possible to deliver the S-SYNC-MAJOR confirm because this would cause V(R) to be updated and this may be a later value than that to which the resynchronize is occurring.

The SPM now waits for an S-RESYNCHRONIZE response.

If a resynchronization contention situation has occurred, only the contention loser (see 7.24.2.3) passes an S-RESYNCHRONIZE indication to the SS-user.

7.24.2.3 Resynchronization contention without symmetric synchronization

The contention between two RESYNCHRONIZE, ACTIVITY INTERRUPT, or ACTIVITY DISCARD SPDUs is resolved according to Table 9. This table defines the contention winner whose SPDU is taken into account; the other SPDU is discarded.

If an incoming RESYNCHRONIZE SPDU is not acceptable, the receiving SS-user may issue another if it prevails over the original proposal according to the decision rules.

Table 9 – Contention winner

OutGoing SPDU from SPM A / Incoming SPDU from SPM B	RESYNCHRONIZE (abandon)	RESYNCHRONIZE (set)	RESYNCHRONIZE (restart)	ACTIVITY INTERRUPT	ACTIVITY DISCARD
RESYNCHRONIZE (abandon)	Initiator	SPM A	SPM A	SPM B	SPM B
RESYNCHRONIZE (set)	SPM B	Initiator	SPM A	SPM B	SPM B
RESYNCHRONIZE (restart)	SPM B	SPM B	SPM with lower serial number or if equal then initiator	SPM B	SPM B
ACTIVITY INTERRUPT	SPM A	SPM A	SPM A	(Note)	(Note)
ACTIVITY DISCARD	SPM A	SPM A	SPM A	(Note)	(Note)

NOTE – Collision is not possible in these cases because only the owner of the major/activity token is permitted to send the ACTIVITY DISCARD SPDU or the ACTIVITY INTERRUPT SPDU.

7.24.3 Sending/receiving the RESYNCHRONIZE SPDU with symmetric synchronization

This subclause defines how the RESYNCHRONIZE SPDU is sent and received when the symmetric synchronize functional unit has been selected.

7.24.3.1 Sending the RESYNCHRONIZE SPDU with symmetric synchronization

In an S-RESYNCHRONIZE request, the SS-user may specify one or both directions of flow. An S-RESYNCHRONIZE request results in a RESYNCHRONIZE SPDU.

If the S-RESYNCHRONIZE request specifies the sending flow, then the First Resync Type and the First Serial Number parameters are present in the RESYNCHRONIZE SPDU. If the First Resync is restart, then the First Serial Number must be greater than or equal to V(Rs) and less than or equal to V(Ms).

If the S-RESYNCHRONIZE request specifies the receiving flow, then the Second Resync and the Second Serial Number parameters are present in the RESYNCHRONIZE SPDU. If the Second Resync is restart, then the Second Serial Number must be greater than or equal to V(Rr) and less than or equal to V(Mr).

The RESYNCHRONIZE SPDU is sent on the transport normal flow. If the transport expedited flow is available to this session connection, a PREPARE (RESYNCHRONIZE) SPDU is sent simultaneously, or earlier, on the transport expedited flow. The PREPARE (RESYNCHRONIZE) SPDU carries the Resync Type for each direction of flow that is to be resynchronized. The Resync Type(s) in the PREPARE (RESYNCHRONIZE) are identical to the one(s) in the RESYNCHRONIZE SPDU.

If resynchronization has been requested only for the receiving flow, then the SPM goes into a state where all the incoming SPDUs are discarded except MINOR SYNC ACK, PREPARE (RESYNCHRONIZE), RESYNCHRONIZE, PREPARE (RESYNCHRONIZE ACK), RESYNCHRONIZE ACK, ACTIVITY DISCARD, ACTIVITY INTERRUPT and ABORT SPDUs. If resynchronization has also been requested for the sending flow, the SPM discards incoming MINOR SYNC ACK SPDUs as well. If resynchronization has been requested for only the sending flow, the SPM goes into the same state but only incoming MINOR SYNC ACK SPDUs are discarded. If a RESYNCHRONIZE, PREPARE (RESYNCHRONIZE), ACTIVITY INTERRUPT or ACTIVITY DISCARD SPDU is received when the SPM is in this state, a resynchronization contention situation has occurred and is dealt with as specified in 7.24.3.3.

7.24.3.2 Receiving the RESYNCHRONIZE SPDU with symmetric synchronization

When the symmetric synchronize functional unit is selected and transport expedited is available, two successive SPDUs will be received:

a) the PREPARE (RESYNCHRONIZE) SPDU on the transport expedited flow, followed by

b) RESYNCHRONIZE SPDU on the transport normal flow.

In this case, the PREPARE SPDU will carry the Resync Type for each direction of flow for which resynchronization is requested.

If the First Resync Type is not present but the Second Resync Type is present in the received PREPARE (RESYNCHRONIZE) SPDU, then resynchronization has not been requested on the requestor's sending flow. In this case, DATA, TYPED DATA and MINOR SYNC POINT SPDUs are processed normally and the requestor's sending flow is not disturbed. MINOR SYNC ACK SPDUs are discarded.

If the First Resync Type is present, then resynchronization has been requested for the requestor's sending flow. In this case, all SPDUs received after the PREPARE (RESYNCHRONIZE) SPDU, except the MINOR SYNC ACK and ABORT SPDUs, are discarded until the RESYNCHRONIZE SPDU is received on the transport normal flow. MINOR SYNC ACK SPDUs are discarded if the Second Resync Type is also present.

Note that if a PREPARE (RESYNCHRONIZE) SPDU is received with neither Resync Type present and the symmetric synchronize functional unit is selected, the PREPARE precedes an ACTIVITY INTERRUPT or ACTIVITY DISCARD SPDU and not a RESYNCHRONIZE SPDU. Both flows are affected.

Except when a resynchronization contention situation has occurred, a valid incoming RESYNCHRONIZE SPDU results in an S-RESYNCHRONIZE indication.

If the First Resync Type is present and the option is restart, then the First Serial Number must be greater than or equal to V(Rr) and less than or equal to V(Mr). If the Second Resync Type is present and the option is restart, then the Second Serial Number parameter must be greater than or equal to V(Rs) and less than or equal to V(Ms).

If the First Resync Type is present and the option is abandon, the S-RESYNCHRONIZE indication contains a serial number which is equal to V(Mr) or the received first serial number, whichever is higher; V(Mr) is set to this value. If the Second Resync Type is present and the option is abandon, the S-RESYNCHRONIZE indication contains a serial number which is equal to V(Ms) or the received second serial number, whichever is higher; V(Ms) is set to this value.

The SPM now waits for an S-RESYNCHRONIZE response.

If a resynchronization contention situation has occurred, only the contention loser (see 7.24.3.3) passes an S-RESYNCHRONIZE indication to the SS-user.

7.24.3.3 Resynchronization contention with symmetric synchronization

The contention between two RESYNCHRONIZE, ACTIVITY INTERRUPT, or ACTIVITY DISCARD SPDUs is resolved as described in 7.24.2.3 when the symmetric synchronize functional unit is selected except in the case where two RESYNCHRONIZE SPDUs collide. When symmetric synchronization is selected and two RESYNCHRONIZE SPDUs collide, each direction of flow is considered separately.

If neither RESYNCHRONIZE SPDU carries a Resync Type parameter for a direction of flow, then that direction of flow is not affected by the resynchronization.

If only one RESYNCHRONIZE SPDU carries a Resync Type parameter for a direction of flow, then the result for that direction of flow is the Resync Type and associated Serial Number in that SPDU.

If the two RESYNCHRONIZE SPDUs carry different Resync Types for a direction of flow, then the result for that direction of flow is the requested Resync Type (and its associated serial number) listed first in the list below:

a) abandon;

b) set;

c) restart.

If both RESYNCHRONIZE SPDUs offer the same Resync Type for a direction of flow, then the result is that Resync Type, and the serial number is determined as defined below:

d) if both RESYNCHRONIZE SPDUs specify abandon, the result for that direction of flow is the higher of the two serial numbers in the SPDUs;

e) if both RESYNCHRONIZE SPDUs specify set, the result for that direction of flow is the serial number in the SPDU sent by the calling SPM;

f) if both RESYNCHRONIZE SPDUs specify restart, the result for that direction of flow is the lower of the two serial numbers in the SPDUs.

If all results for the affected direction(s) of flow were provided in the RESYNCHRONIZE SPDU sent by one and only one SPM, then that SPM "wins" the contention and the RESYNCHRONIZE SPDU sent by the other SPM is discarded. Otherwise, the Resync Type and Serial Number results replace the appropriate parameters in the RESYNCHRONIZE SPDU sent by the calling SPM. The RESYNCHRONIZE SPDU sent by the called SPM is discarded. A RESYNCHRONIZE indication is issued to the called SS-user.

If an incoming RESYNCHRONIZE indication is not acceptable, the SS-user may issue another if it prevails over the indication according to the decision rules.

If a RESYNCHRONIZE SPDU carrying only a Resync Type parameter for the requestor's sending flow collides with a MAP SPDU, a collision has occurred. In this case, an S-SYNC-MAJOR indication is delivered to the receiving SS-user, but the receiving SS-user is not allowed to respond with an S-SYNC-MAJOR response.

An SS-user is not allowed to request to resynchronize only its sending flow after receiving a major synchronization point indication but before issuing the response.

7.25 RESYNCHRONIZE ACK SPDU

The RESYNCHRONIZE ACK SPDU is used to notify the sender of a RESYNCHRONIZE SPDU of the completion of resynchronization.

7.25.1 Content of RESYNCHRONIZE ACK SPDU

The RESYNCHRONIZE ACK SPDU contains:

a) An Enclosure Item parameter to indicate whether the SPDU is the beginning, middle or end of the SSDU (see also 7.37.1). This parameter shall not be present when Protocol Version 1 has been selected.

b) A Token Setting Item parameter which indicates the selected token positions.

c) A Resync Type parameter indicates the First Resync Type, which is the resynchronize option for the requestor's sending flow in the S-RESYNCHRONIZE indication. This parameter may only be present if the symmetric synchronize functional unit has been selected.

d) A Serial Number parameter which indicates the first serial number to be used in the resynchronized flow.

When the symmetric synchronize functional unit has not been selected, this parameter is set according to the Resync Type parameter in the received RESYNCHRONIZE SPDU:

1) for the restart option, to the serial number in the received RESYNCHRONIZE SPDU;

2) for the set option, to the serial number in the S-RESYNCHRONIZE response;

3) for the abandon option, to V(M).

When the symmetric synchronize functional unit has been selected, this parameter indicates the First Serial Number, which is the first serial number to be used on the sending flow of the requestor of the resynchronize. This parameter and the Second Serial Number parameter are set according to the associated First and Second Resync Type parameter as follows:

4) for the restart option, to the serial number in the S-RESYNCHRONIZE indication;

5) for the set option, to the serial number in the S-RESYNCHRONIZE response;

6) for the abandon option, the First Serial Number is set to the Serial Number parameter received in the RESYNCHRONIZE SPDU or to V(Mr), whichever is higher; the Second Serial Number is set to the Second Serial Number parameter received in the RESYNCHRONIZE SPDU or to V(Ms), whichever is higher.

e) A Second Resync Type parameter which indicates the resynchronize option for the requestor's receiving flow in the S-RESYNCHRONIZE indication. This parameter may only be present if the symmetric synchronize functional unit has been selected.

f) A Second Serial Number parameter which indicates the next serial number to be used on the requestor's receiving flow. This parameter is set as defined in c) above according to the Second Resync Type parameter in the S-RESYNCHRONIZE indication. This parameter may only be present if the symmetric synchronize functional unit has been selected.

g) A User Data parameter which allows transparent user data to be transferred.

7.25.2 Sending/receiving the RESYNCHRONIZE ACK SPDU without symmetric synchronization

This subclause defines how the RESYNCHRONIZE ACK SPDU is sent and received when the symmetric synchronize functional unit has not been selected.

7.25.2.1 Sending the RESYNCHRONIZE ACK SPDU without symmetric synchronization

An S-RESYNCHRONIZE response results in a RESYNCHRONIZE ACK SPDU. This SPDU is sent on the transport normal flow. If the transport expedited flow is available to this session connection, a PREPARE (RESYNCHRONIZE ACK) SPDU is sent simultaneously, or earlier, on the transport expedited flow.

The tokens are set to the values proposed by the requestor. If the requestor has indicated "accepting SS-user's choice" for a token, then the acceptor's proposed value for that token is used. The selected token settings are returned in the Token Setting Item of the RESYNCHRONIZE ACK SPDU.

V(A) and V(M) are set to the serial number contained in the RESYNCHRONIZE ACK SPDU.

V(R) is unchanged if the Resync Type parameter in the received RESYNCHRONIZE SPDU indicated the restart option. Otherwise, V(R) is set to zero.

If the data separation functional unit has been selected, V(Ado) is set to –1.

7.25.2.2 Receiving the RESYNCHRONIZE ACK SPDU without symmetric synchronization

A valid incoming RESYNCHRONIZE ACK SPDU results in an S-RESYNCHRONIZE confirm. If the transport expedited flow is available to this session connection, two successive SPDUs will be received:

a) PREPARE (RESYNCHRONIZE ACK) SPDU on the transport expedited flow, followed by

b) RESYNCHRONIZE ACK on the transport normal flow.

The tokens are set to the positions specified in the RESYNCHRONIZE ACK SPDU.

V(A) and V(M) are set to the serial number contained in the RESYNCHRONIZE ACK SPDU.

V(R) is unchanged if the Resync Type parameter in the transmitted RESYNCHRONIZE SPDU indicated the restart option. Otherwise, V(R) is set to zero.

If the data separation functional unit has been selected, V(Ado) is set to –1.

7.25.3 Sending/receiving the RESYNCHRONIZE ACK SPDU with symmetric synchronization

This subclause defines how the RESYNCHRONIZE ACK SPDU is sent and received when the symmetric synchronize functional unit has been selected.

7.25.3.1 Sending the RESYNCHRONIZE ACK SPDU with symmetric synchronization

An S-RESYNCHRONIZE response results in a RESYNCHRONIZE ACK SPDU. This SPDU is sent on the transport normal flow. If the transport expedited flow is available to this session connection, a PREPARE (RESYNCHRONIZE ACK) SPDU is sent simultaneously, or earlier, on the transport expedited flow.

The tokens are set to the values proposed by the requestor. If the requestor has indicated "accepting SS-user's choice" for a token, then the acceptor's proposed value for that token is used. The selected token settings are returned in the Token Setting Item of the RESYNCHRONIZE ACK SPDU.

V(Ar) and V(Mr) are set to the First Serial Number parameter, if present in the RESYNCHRONIZE ACK SPDU. V(As) and V(Ms) are set to the Second Serial Number parameter, if present in the RESYNCHRONIZE ACK SPDU.

V(Rr) is unchanged if the First Resync Type parameter in the RESYNCHRONIZE ACK SPDU is not present or indicates the restart option. Otherwise, V(Rr) is set to zero.

V(Rs) is unchanged if the Second Resync Type parameter in the RESYNCHRONIZE ACK SPDU is not present or indicates the restart option. Otherwise, V(Rs) is set to zero.

7.25.3.2 Receiving the RESYNCHRONIZE ACK SPDU with symmetric synchronization

A valid incoming RESYNCHRONIZE ACK SPDU results in an S-RESYNCHRONIZE confirm. If the transport expedited flow is available to this session connection, two successive SPDUs will be received:

a) PREPARE (RESYNCHRONIZE ACK) SPDU on the transport expedited flow, followed by

b) RESYNCHRONIZE ACK on the transport normal flow.

The tokens are set to the positions in the RESYNCHRONIZE ACK SPDU.

V(As) and V(Ms) are set to the First Serial Number parameter, if present in the RESYNCHRONIZE ACK SPDU. V(Ar) and V(Mr) are set to the Second Serial Number parameter, if present in the RESYNCHRONIZE ACK SPDU.

V(Rr) is unchanged if the First Resync Type parameter in the RESYNCHRONIZE ACK SPDU is not present or indicates the restart option. Otherwise, V(Rr) is set to zero.

V(Rs) is unchanged if the Second Resync Type parameter in the RESYNCHRONIZE ACK SPDU is not present or indicates the restart option. Otherwise, V(Rs) is set to zero.

7.26 PREPARE SPDU

The PREPARE SPDU is only used when the transport expedited flow is available to this session connection. It notifies the imminent arrival of certain SPDUs and indicate to the receiving SPM that SPDUs received on the transport normal flow may be discarded under certain circumstances.

7.26.1 Content of PREPARE SPDU

The PREPARE SPDU contains a Prepare Type parameter which indicates which SPDU should be expected on the transport normal flow.

When the PREPARE SPDU is used to notify the imminent arrival of a RESYNCHRONIZE SPDU and the symmetric synchronize functional unit is selected, the PREPARE SPDU contains:

a) A Prepare Type parameter indicating RESYNCHRONIZE.

b) A First Resync Type parameter, which is present only if this parameter is present in the RESYNCHRONIZE SPDU. Its value is identical to that of the First Resync Type parameter in the RESYNCHRONIZE SPDU.

c) A Second Resync Type parameter, which is present only if this parameter is present in the RESYNCHRONIZE SPDU. Its value is identical to that of the Second Resync Type parameter in the RESYNCHRONIZE SPDU.

7.26.2 Sending the PREPARE SPDU

The PREPARE SPDU is sent before the associated SPDUs specified in Table 10 when the transport expedited flow is available to this session connection. Table 10 also specifies the value of the Prepare Type parameter.

The PREPARE SPDU is sent on the transport expedited flow (its associated SPDU being sent on the transport normal flow). The SPM goes to a state which is determined by the initial request.

Table 10 – SPDUs associated with the PREPARE SPDU

Associated SPDU	Prepare Type
RESYNCHRONIZE SPDU	RESYNCHRONIZE
RESYNCHRONIZE ACK SPDU	RESYNCHRONIZE ACK
MAJOR SYNC ACK SPDU	MAJOR SYNC ACK
ACTIVITY INTERRUPT SPDU	RESYNCHRONIZE
ACTIVITY INTERRUPT ACK SPDU	RESYNCHRONIZE ACK
ACTIVITY DISCARD SPDU	RESYNCHRONIZE
ACTIVITY DISCARD ACK SPDU	RESYNCHRONIZE ACK
ACTIVITY END ACK SPDU	MAJOR SYNC ACK
ABORT SPDU	ABORT

7.26.3 Receiving the PREPARE SPDU

A valid incoming PREPARE SPDU results in the SPM entering a state where it is waiting for the associated SPDU on the transport normal flow. If the Prepare Type parameter indicates MAJOR SYNC ACK, any SPDUs received on the

transport normal flow are processed normally. If the Prepare Type parameter indicates RESYNCHRONIZE and the symmetric synchronize functional unit is selected, then processing occurs as follows:

a) If neither Resync Type parameter is present, then SPDUs received on the transport normal flow are discarded until an ACTIVITY INTERRUPT or ACTIVITY DISCARD SPDU is received.

b) If the First Resync Type parameter is not present but the Second Resync Type parameter is present, then DATA, TYPED DATA and MINOR SYNC POINT SPDUs are processed normally. MINOR SYNC POINT SPDUs are discarded.

c) If the First Resync Type parameter is present, then all SPDUs received on the transport normal flow are discarded (except for MINOR SYNC ACK and ABORT) until the RESYNCHRONIZE SPDU is received. MINOR SYNC ACK SPDUs are also discarded if the Second Resync Type parameter is also present.

Otherwise, SPDUs received on the transport normal flow before the indicated SPDU are discarded. If an EXPEDITED DATA SPDU is validly received after a PREPARE SPDU, but before the associated SPDU on the transport normal flow, the S-EXPEDITED-DATA indication is not passed to the SS-user until the associated SPDU has been received and processed.

7.27 EXCEPTION REPORT SPDU

The EXCEPTION REPORT SPDU is used to report that a protocol error has been detected within the SPM. It can only be sent in the data transfer phase and subject to the token restrictions specified in Table 5.

7.27.1 Content of EXCEPTION REPORT SPDU

The EXCEPTION REPORT SPDU contains a Reflect Parameter Values parameter which is used to indicate a field of arbitrary length, which contains the bit pattern of the SPDU received with a protocol error, up to and including the detected error.

7.27.2 Sending the EXCEPTION REPORT SPDU

On detection of a protocol error, for example an SPDU received at an unexpected time, or an invalid SPDU, the SPM may generate an EXCEPTION REPORT SPDU. This SPDU is sent on the transport normal flow. At the same time an S-P-EXCEPTION-REPORT indication will be generated. The SPM enters an error state which is only left when any of the following SPDUs, or their associated local service requests, are received:

- ACTIVITY DISCARD;
- ACTIVITY INTERRUPT;
- RESYNCHRONIZE;
- ABORT;
- GIVE TOKENS (with the data token);
- PREPARE (RESYNCHRONIZE).

Any other SPDUs received will be discarded. However, V(A) and V(M) will be updated appropriately if valid MINOR SYNC POINT SPDUs or MAJOR SYNC POINT SPDUs are received.

7.27.3 Receiving the EXCEPTION REPORT SPDU

When an incoming EXCEPTION REPORT SPDU is received, an S-P-EXCEPTION-REPORT indication is given and the SPM enters an error state.

The SPM leaves the error state when any of the following SPDUs, or their associated local service requests, are received:

- ACTIVITY DISCARD;
- ACTIVITY INTERRUPT;
- RESYNCHRONIZE;
- ABORT;
- GIVE TOKENS (with the data token);
- PREPARE (RESYNCHRONIZE).

NOTE – This action is dependent on the receipt of the EXCEPTION REPORT SPDU, not on examination of its parameter value. This enables the procedure to be followed in cases where the implementation cannot deal with an SPDU length greater than the minimum specified in 8.3.27.3.

7.28 EXCEPTION DATA SPDU

The EXCEPTION DATA SPDU is used to put the SPM into an error state.

It can only be sent subject to the token restrictions specified in Table 5 and:

a) when the activity management functional unit has been selected and an activity is in progress; or

b) the activity management functional unit has not been selected.

7.28.1 Content of EXCEPTION DATA SPDU

The EXCEPTION DATA SPDU contains:

a) An Enclosure Item parameter to indicate whether the SPDU is the beginning, middle or end of the SSDU (see also 7.37.1). This parameter shall not be present when Protocol Version 1 has been selected.

b) A Reason Code parameter which indicates the reason for sending the EXCEPTION DATA SPDU.

c) A User Data parameter which allows transparent user data to be transferred.

7.28.2 Sending the EXCEPTION DATA SPDU

An S-U-EXCEPTION-REPORT request results in the SPM sending an EXCEPTION DATA SPDU on the transport normal flow. The SPM enters an error state. The error state will be left when an S-U-ABORT request or a T-DISCONNECT indication is received or when any of the following SPDUs are received:

- ACTIVITY DISCARD;
- ACTIVITY INTERRUPT;
- RESYNCHRONIZE;
- ABORT;
- GIVE TOKENS (with the data token);
- PREPARE (RESYNCHRONIZE).

Any other SPDUs received will be discarded. However, V(A) and V(M) will be updated appropriately if MINOR SYNC POINT SPDUs or MAJOR SYNC POINT SPDUs are received.

7.28.3 Receiving the EXCEPTION DATA SPDU

A valid incoming EXCEPTION DATA SPDU results in an S-U-EXCEPTION-REPORT indication. The SPM enters an error state, unless the data token is not assigned to this SPM, in which case the SPM state is unchanged.

The SPM leaves the error state when any of the following service primitives are invoked by the SS-user:

- S-U-ABORT request;
- S-RESYNCHRONIZE request;
- S-ACTIVITY-DISCARD request;
- S-ACTIVITY-INTERRUPT request;
- S-TOKEN-GIVE request (with the data token).

7.29 ACTIVITY START SPDU

The ACTIVITY START SPDU is used to notify the beginning of an activity. The right to issue an ACTIVITY START SPDU is subject to the token restrictions specified in Table 5.

7.29.1 Content of ACTIVITY START SPDU

The ACTIVITY START SPDU contains:

a) An Enclosure Item parameter to indicate whether the SPDU is the beginning, middle or end of the SSDU (see also 7.37.1). This parameter shall not be present when Protocol Version 1 has been selected.

b) An Activity Identifier parameter which allows the SS-users to identify the activity being started.

c) A User Data parameter which allows transparent user data to be transferred.

7.29.2 Sending the ACTIVITY START SPDU

An S-ACTIVITY-START request (when Vact is false) results in an ACTIVITY START SPDU. Vact is set true. If the symmetric synchronize functional unit has been selected, V(As), V(Ar), V(Ms), V(Mr), V(Rs), and V(Rr) are set to one. Otherwise, V(A), V(M), and V(R) are set to one. This SPDU is sent on the transport normal flow.

7.29.3 Receiving the ACTIVITY START SPDU

A valid incoming ACTIVITY START SPDU (when Vact is false) results in an S-ACTIVITY-START indication. Vact is set true. If the symmetric synchronize functional unit has been selected, V(As), V(Ar), V(Ms), V(Mr), V(Rs), and V(Rr) are set to one. Otherwise, V(A), V(M), and V(R) are set to one.

7.30 ACTIVITY RESUME SPDU

The ACTIVITY RESUME SPDU is used to notify the resumption of a previously interrupted activity. The right to issue an ACTIVITY RESUME SPDU is subject to the token restrictions specified in Table 5.

7.30.1 Content of ACTIVITY RESUME SPDU

The ACTIVITY RESUME SPDU contains:

a) An Enclosure Item parameter to indicate whether the SPDU is the beginning, middle or end of the SSDU (see also 7.37.1). This parameter shall not be present when Protocol Version 1 has been selected.

b) A Linking Information parameter group which contains:

1) A Called SS-user Reference parameter.

2) A Calling SS-user Reference parameter.

3) A Common Reference parameter.

4) An Additional Reference Information parameter.

5) An Old Activity Identifier which enables the SS-users to identify the old activity which is being resumed.

6) A Serial Number parameter which indicates the first serial number minus one. When the symmetric synchronize function unit has been selected, the parameter indicates the First Serial Number parameter, which is the serial number provided by the requesting SS-user for use on the sending data flow of the requestor of the activity resume.

7) A Second Serial Number parameter which indicates the serial number provided by the requesting SS-user for use on the requestor's receiving data flow. This parameter is present only if the symmetric synchronize functional unit has been selected.

c) A New Activity Identifier parameter which allows the SS-users to assign a new identifier to the activity being resumed.

d) A User Data parameter which allows transparent user data to be transferred.

7.30.2 Sending the ACTIVITY RESUME SPDU

An S-ACTIVITY-RESUME request (when Vact is false) results in an ACTIVITY RESUME SPDU. Vact is set true. If the symmetric synchronize functional unit has been selected, V(As) and V(Ms) are set to the First Serial Number parameter provided by the SS-user plus one; V(Ar) and V(Mr) are set to the Second Serial Number parameter provided by the SS-user plus one; V(Rs) and V(Rr) are set to one. Otherwise, V(A) and V(M) are set to the serial number provided by the SS-user plus one; V(R) is set to one. This SPDU is sent on the transport normal flow.

7.30.3 Receiving the ACTIVITY RESUME SPDU

A valid incoming ACTIVITY RESUME SPDU (when Vact is false) results in an S-ACTIVITY-RESUME indication. Vact is set true. If the symmetric synchronize functional unit has been selected, V(Ar) and V(Mr) are set to the received First Serial Number parameter provided by the SS-user plus one; V(As) and V(Ms) are set to the received Second Serial Number parameter plus one; V(Rs) and V(Rr) are set to one. Otherwise, V(A) and V(M) are set to the received serial number plus one; V(R) is set to one.

7.31 ACTIVITY INTERRUPT SPDU

The ACTIVITY INTERRUPT SPDU is used to notify the interruption of an ongoing activity. The right to issue an ACTIVITY INTERRUPT SPDU is subject to the token restrictions specified in Table 5. Use of this procedure may result in loss of data.

7.31.1 Content of ACTIVITY INTERRUPT SPDU

The ACTIVITY INTERRUPT SPDU contains:

a) An Enclosure Item parameter to indicate whether the SPDU is the beginning, middle or end of the SSDU (see also 7.37.1). This parameter shall not be present when Protocol Version 1 has been selected.

b) A Reason Code parameter which indicates the reason for sending the ACTIVITY INTERRUPT SPDU.

c) A User Data parameter which allows transparent user data to be transferred. This parameter shall not be present if Protocol Version 1 is selected.

7.31.2 Sending the ACTIVITY INTERRUPT SPDU

An S-ACTIVITY-INTERRUPT request results in an ACTIVITY INTERRUPT SPDU. This SPDU is sent on the transport normal flow. If the transport expedited flow is available to this session connection, a PREPARE (RESYNCHRONIZE) SPDU is sent simultaneously, or earlier, on the transport expedited flow. The SPM goes into a state where all incoming SPDUs are discarded except PREPARE (RESYNCHRONIZE ACK), ACTIVITY INTERRUPT ACK and ABORT.

7.31.3 Receiving the ACTIVITY INTERRUPT SPDU

A valid incoming ACTIVITY INTERRUPT SPDU results in an S-ACTIVITY-INTERRUPT indication. If the transport expedited flow is available to this session connection, two successive SPDUs will be received:

a) PREPARE (RESYNCHRONIZE) SPDU (see 7.24) on the transport expedited flow, followed by

b) ACTIVITY INTERRUPT SPDU on the transport normal flow.

The SPM now waits for an S-ACTIVITY-INTERRUPT response.

7.32 ACTIVITY INTERRUPT ACK SPDU

The ACTIVITY INTERRUPT ACK SPDU is used to notify the sender of an ACTIVITY INTERRUPT SPDU of the completion of the interruption of the ongoing activity. On completion, all available tokens are assigned to the sender of the ACTIVITY INTERRUPT SPDU.

7.32.1 Content of ACTIVITY INTERRUPT ACK SPDU

The ACTIVITY INTERRUPT ACK SPDU contains:

a) An Enclosure Item parameter to indicate whether the SPDU is the beginning, middle or end of the SSDU (see also 7.37.1). This parameter shall not be present when Protocol Version 1 has been selected.

b) A User Data parameter which allows transparent user data to be transferred. This parameter shall not be present if Protocol Version 1 is selected.

7.32.2 Sending the ACTIVITY INTERRUPT ACK SPDU

An S-ACTIVITY-INTERRUPT response results in an ACTIVITY INTERRUPT ACK SPDU. This SPDU is sent on the transport normal flow. If the transport expedited flow is available to this session connection, a PREPARE (RESYNCHRONIZE ACK) SPDU is sent simultaneously, or earlier, on the transport expedited flow. Vact is set false when the ACTIVITY INTERRUPT ACK SPDU has been sent.

7.32.3 Receiving the ACTIVITY INTERRUPT ACK SPDU

A valid incoming ACTIVITY INTERRUPT ACK SPDU results in an S-ACTIVITY-INTERRUPT confirm. If the transport expedited flow is available to this session connection, two successive SPDUs will be received:

a) PREPARE (RESYNCHRONIZE ACK) SPDU (see 7.25) on the transport expedited flow, followed by

b) ACTIVITY INTERRUPT ACK SPDU on the transport normal flow.

Vact is set false when the ACTIVITY INTERRUPT ACK SPDU has been received.

7.33 ACTIVITY DISCARD SPDU

The ACTIVITY DISCARD SPDU is used to notify the cancellation of an ongoing activity. The right to issue an ACTIVITY DISCARD SPDU is subject to the token restrictions specified in Table 5. Use of this procedure may result in the loss of data.

7.33.1 Content of ACTIVITY DISCARD SPDU

The ACTIVITY DISCARD SPDU contains:

a) An Enclosure Item parameter to indicate whether the SPDU is the beginning, middle or end of the SSDU (see also 7.37.1). This parameter shall not be present when Protocol Version 1 has been selected.

b) A Reason Code parameter which indicates the reason for sending the ACTIVITY DISCARD SPDU.

c) A User Data parameter which allows transparent user data to be transferred. This parameter shall not be present if Protocol Version 1 is selected.

7.33.2 Sending the ACTIVITY DISCARD SPDU

An S-ACTIVITY-DISCARD request results in an ACTIVITY DISCARD SPDU. This SPDU is sent on the transport normal flow. If the transport expedited flow is available to this session connection, a PREPARE (RESYNCHRONIZE) SPDU is sent simultaneously, or earlier, on the transport expedited flow. The SPM goes into a state where all the incoming SPDUs are discarded except PREPARE (RESYNCHRONIZE ACK), ACTIVITY DISCARD ACK and ABORT SPDUs.

7.33.3 Receiving the ACTIVITY DISCARD SPDU

A valid incoming ACTIVITY DISCARD SPDU results in an S-ACTIVITY-DISCARD indication. If the transport expedited flow is available to this session connection, two successive SPDUs will be received:

a) PREPARE (RESYNCHRONIZE) SPDU (see 7.24) on the transport expedited flow, followed by

b) ACTIVITY DISCARD SPDU on the transport normal flow.

The SPM now waits for an S-ACTIVITY-DISCARD response.

7.34 ACTIVITY DISCARD ACK SPDU

The ACTIVITY DISCARD ACK SPDU is used to notify the sender of an ACTIVITY DISCARD SPDU of the completion of the cancellation of the ongoing activity. On completion, all available tokens are assigned to the sender of the ACTIVITY DISCARD SPDU.

7.34.1 Content of ACTIVITY DISCARD ACK SPDU

The ACTIVITY DISCARD ACK SPDU contains:

a) An Enclosure Item parameter to indicate whether the SPDU is the beginning, middle or end of the SSDU (see also 7.37.1). This parameter shall not be present when Protocol Version 1 has been selected.

b) A User Data parameter which allows transparent user data to be transferred. This parameter shall not be present if Protocol Version 1 is selected.

7.34.2 Sending the ACTIVITY DISCARD ACK SPDU

An S-ACTIVITY-DISCARD response results in an ACTIVITY DISCARD ACK SPDU. This SPDU is sent on the transport normal flow. If the transport expedited flow is available to this session connection, a PREPARE (RESYNCHRONIZE ACK) SPDU is sent simultaneously, or earlier, on the transport expedited flow. Vact is set false when the ACTIVITY DISCARD ACK SPDU has been sent.

7.34.3 Receiving the ACTIVITY DISCARD ACK SPDU

A valid incoming ACTIVITY DISCARD ACK SPDU results in an S-ACTIVITY-DISCARD confirm. If the transport expedited flow is available to this session connection, two successive SPDUs will be received:

a) PREPARE (RESYNCHRONIZE ACK) SPDU (see 7.25) on the transport expedited flow, followed by

b) ACTIVITY DISCARD ACK SPDU on the transport normal flow.

Vact is set false when the ACTIVITY DISCARD ACK SPDU has been received.

7.35 ACTIVITY END SPDU

The ACTIVITY END SPDU is used to define an implicit major synchronization point at the end of an activity. A confirmation has to be received before more data can be sent on the normal and expedited flows. The right to issue an ACTIVITY END SPDU is subject to the token restrictions specified in Table 5.

An ACTIVITY END SPDU can only be validly sent when Vact is true.

7.35.1 Content of ACTIVITY END SPDU

The ACTIVITY END SPDU contains:

a) An Enclosure Item parameter to indicate whether the SPDU is the beginning, middle or end of the SSDU (see also 7.37.1). This parameter shall not be present when Protocol Version 1 has been selected.

b) A Serial Number parameter which indicates the serial number of this major synchronization point. When the symmetric synchronize functional unit has been selected, this parameter indicates the First Serial Number, which is associated with the sending flow of the requestor of the major synchronization point, and is set by the sending SPM to the current value of V(Ms). It is set to the current value of V(M when the symmetric synchronize functional unit has not been selected.

c) A User Data parameter which allows transparent user data to be transferred.

7.35.2 Sending the ACTIVITY END SPDU

An S-ACTIVITY-END request (when Vact is true) results in an ACTIVITY END SPDU. This SPDU is sent on the transport normal flow. Vnextact is set false. If the symmetric synchronize functional unit has been selected, V(Ms) is incremented by one. Otherwise, if Vsc is true, V(A) is set equal to V(M) and Vsc is set false; V(M) is incremented by one. If the transport expedited flow is available to this session connection, the SPM waits for a PREPARE (MAJOR SYNC ACK) SPDU, followed by an ACTIVITY END ACK SPDU. Otherwise, just an ACTIVITY END ACK SPDU is expected. Any other SPDUs received prior to the ACTIVITY END ACK SPDU will result in the appropriate service indications being given to the SS-user.

7.35.3 Receiving the ACTIVITY END SPDU

A valid incoming ACTIVITY END SPDU results in an S-ACTIVITY-END indication. Vact must be true. If the symmetric synchronize functional unit has been selected, then the received serial number must equal V(Mr); V(Mr) is incremented by one, and Vnextact is set false. Otherwise, the received serial number must equal V(M); if Vsc is false, V(A) is set equal to V(M); V(M) is incremented by one; Vnextact is set false.

- GIVE TOKENS CONFIRM SPDU;
- MINOR SYNC POINT SPDU;
- MINOR SYNC ACK SPDU;
- MAJOR SYNC POINT SPDU;
- MAJOR SYNC ACK SPDU;
- RESYNCHRONIZE SPDU;
- RESYNCHRONIZE ACK SPDU;
- EXCEPTION DATA SPDU;
- ACTIVITY START SPDU;
- ACTIVITY RESUME SPDU;
- ACTIVITY INTERRUPT SPDU;
- ACTIVITY INTERRUPT ACK SPDU;
- ACTIVITY DISCARD SPDU;
- ACTIVITY DISCARD ACK SPDU;
- ACTIVITY END SPDU;
- ACTIVITY END ACK SPDU.

These SPDUs are subject to the following additional procedures.

7.37.1 Content of the SPDU

Where an SSDU is segmented, the first SPDU contains all the parameters which would have been present in the SPDU if the SSDU had not been segmented, together with an Enclosure Item parameter, which indicates beginning of SSDU and not end of SSDU, and at least one octet of User Data. The last SPDU of the SSDU contains the Enclosure Item parameter which indicates not beginning of SSDU and end of SSDU and may or may not contain User Data. Intermediate SPDUs of the SSDU, if present, contain the Enclosure Item parameter which indicates not beginning of SSDU and not end of SSDU and at least one octet of User Data.

7.37.2 Sending the SPDU

The sending procedures for SPDUs where these additional elements of procedure apply are extended in the following way:

a) Where the SPM sends an SPDU, it shall send an ordered sequence of SPDUs which together comprise the complete SSDU.

b) Sending this ordered sequence of SPDUs shall be interrupted when the SPM sends an ABORT SPDU or a PREPARE (ABORT) SPDU (for example, as a result of an S-U-ABORT request or a detected protocol error) or when the SPM receives an ABORT SPDU, a PREPARE (ABORT) SPDU or a T-DISCONNECT indication. In this case, the SPM shall stop sending the ordered sequence of SPDUs and shall take the appropriate defined actions.

NOTE 1 – the ordered sequence of SPDUs sent so far will not comprise a complete SSDU. The Enclosure Item parameter will not have been sent with a value which indicates end of SSDU.

c) As a local matter, sending this ordered sequence of SPDUs may be interrupted when the SPM receives an SPDU which will cause the ordered sequence of SPDUs to be discarded by the remote SPM. In this situation, the SPM which is sending the ordered sequence of SPDUs is not required to send the remainder of the ordered sequence.

NOTE 2 – This situation will occur if the received destructive SPDU was sent by the remote SPM before the first SPDU of the ordered sequence of SPDUs was received by the remote SPM, or if the remote SPM took the local implementation decision indicated in 7.37.3 d).

7.37.3 Receiving the SPDU

The receiving procedures for SPDUs where these additional elements of procedure apply are extended in the following way:

a) Where the SPM receives an SPDU, it shall receive an ordered sequence of SPDUs which together comprise the complete SSDU.

7.36 ACTIVITY END ACK SPDU

The ACTIVITY END ACK SPDU is used to return a confirmation to an ACTIVITY END SPDU.

7.36.1 Content of ACTIVITY END ACK SPDU

The ACTIVITY END ACK SPDU contains:

a) An Enclosure Item parameter to indicate whether the SPDU is the beginning, middle or end of the SSDU (see also 7.37.1). This parameter shall not be present when Protocol Version 1 has been selected.

b) A Serial Number parameter which indicates the serial number of the major synchronization point which is being confirmed [which is equal to V(M) minus one]. This parameter is only present with the symmetric synchronize functional unit has not been selected.

c) A Second Serial Number parameter which indicates the serial number of the major synchronization point for the receiving flow of the requestor of the major synchronization point, which is the direction of flow on which the MAJOR SYNC ACK SPDU travels. This number is set by the SPM to the current value of V(Ms). This parameter is only present when the symmetric synchronize functional unit has been selected.

d) A User Data parameter which allows transparent user data to be transferred.

7.36.2 Sending the ACTIVITY END ACK SPDU

An S-ACTIVITY-END response results in an ACTIVITY END ACK SPDU. This SPDU is sent on the transport normal flow. If the transport expedited flow is available to this session connection, a PREPARE (MAJOR SYNC ACK) SPDU is sent simultaneously, or earlier, on the transport expedited flow. Vact is set to Vnextact. If the symmetric synchronize functional unit has been selected, V(Ms) is incremented by one; V(As) and V(Rs) are set equal to V(Ms); and V(Ar) and V(Rr) are set equal to V(Mr). Otherwise V(A) and V(R) are set equal to V(M).

7.36.3 Receiving the ACTIVITY END ACK SPDU

A valid incoming ACTIVITY END ACK SPDU results in an S-ACTIVITY-END confirm.

If the symmetric synchronize functional unit has been selected, the Second Serial Number is equal to V(Mr); V(Mr) is incremented by one; V(As) and V(Rs) are set equal to V(Ms); V(Ar) and V(Rr) are set equal to V(Mr); Vact is set to Vnextact.

Otherwise, Vsc must be false and the Serial Number must be equal to V(M) minus one; V(A) and V(R) are set equal to V(M); Vact is set to Vnextact.

If the transport expedited flow is available to this session connection, two successive SPDUs will be received:

a) PREPARE (MAJOR SYNC ACK) SPDU on the transport expedited flow, followed by

b) ACTIVITY END ACK SPDU on the transport normal flow.

V(A) and V(R) are set equal to V(M). Vact is set to Vnextact.

7.37 Additional Elements of Procedure for Segmented SSDUs

The following SPDUs may contain segments of the associated SSDU:

- ACCEPT SPDU;
- REFUSE SPDU;
- FINISH SPDU;
- DISCONNECT SPDU;
- NOT FINISHED SPDU;
- CAPABILITY DATA SPDU;
- CAPABILITY DATA ACK SPDU;
- GIVE TOKENS SPDU;
- PLEASE TOKENS SPDU;

b) Receiving this ordered sequence of SPDUs shall be interrupted when the SPM receives an ABORT SPDU, a PREPARE (ABORT) SPDU or a T-DISCONNECT indication. This shall have a destructive effect on the entire segmented SSDU (i.e. the SPDUs which have already been received are discarded). The SPM shall take the appropriate defined actions.

c) The SPM may send an ABORT SPDU or a PREPARE (ABORT) SPDU (for example, as a result of an S-U-ABORT request or a detected protocol error) while receiving an ordered sequence of SPDUs. This shall have a destructive effect on the entire segmented SSDU being received (i.e. SPDUs comprising part of the segmented SSDU which have already been received are discarded and any SPDUs comprising part of the segmented SSDU which are received subsequently are discarded). The SPM shall take the appropriate defined actions.

d) As a local matter, while receiving this ordered sequence of SPDUs, the SPM may send any other appropriate SPDU which will have a destructive effect on the entire SSDU being received (i.e. SPDUs comprising part of the segmented SSDU which have already been received will be discarded and any SPDUs comprising part of the segmented SSDU which are received subsequently will be discarded).

NOTE – The conditions and effect stated here are as though the segmented SSDU had been sent in a single SPDU and the SPDU causing the destructive effect had been sent before that SPDU is received.

8 Structure and encoding of SPDUs

8.1 TSDU structure

Each TSDU consists of one or more SPDUs complying with the requirements for concatenation (see 6.3.7).

Each SPDU within a TSDU consists of one or more octets that are numbered sequentially starting from 1.

Each octet within an SPDU consists of eight bits numbered 8 to 1 where 1 is the low ordered bit.

The sequence of octets within an SPDU and the sequence of bits within an octet are defined for each SPDU in 8.3, with the additional convention that where the text refers to bits within a two octet field and the bits are numbered 16 to 1, then 1 is the low order bit and the octet containing bits 16 to 9 precedes the octet containing bits 8 to 1 in the SPDU.

Within each TSDU:

a) the sequential ordering of SPDUs is maintained;

b) the ordering of the octets is maintained in the same order as in the SPDU;

c) the ordering of bits within each TSDU is maintained in the same order as in the SPDU (i.e. the low order bit is mapped onto the low order bit and the high order bit is mapped onto the high order bit).

NOTES

1 The TSDU structure is illustrated in Figure 3. The integrity of this structure is maintained over a transport connection. This Recommendation | International Standard does not define the way in which the TSDU is transmitted.

2 When the structure of an SPDU is illustrated in this Recommendation | International Standard, the following convention is used:

a) octets are shown with the lowest numbered octet to the left, higher numbered octets being shown further to the right;

b) within an octet, bits are shown with bit 8 to the left and bit 1 to the right.

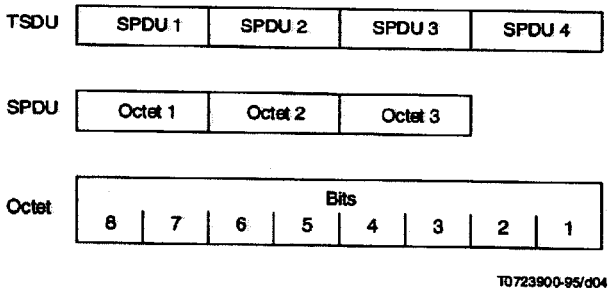

Figure 3 – Illustration of definition of TSDU structure

8.2 SPDU structure

This subclause specifies the general structure of SPDUs in terms of their constituent fields. This structure is illustrated in Figure 4.

Codings and structural requirements specific to particular SPDUs are specified in 8.3.

Examples of valid SPDU structure are illustrated in Figure 5.

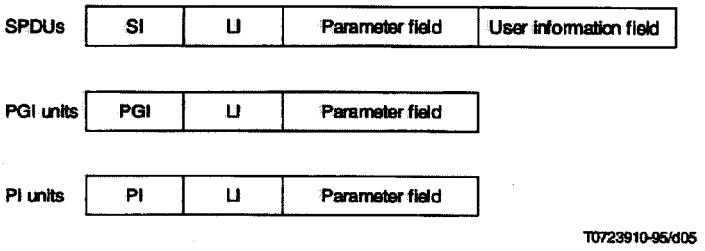

Figure 4 – Illustration of structure of SPDUs, PGI and PI units

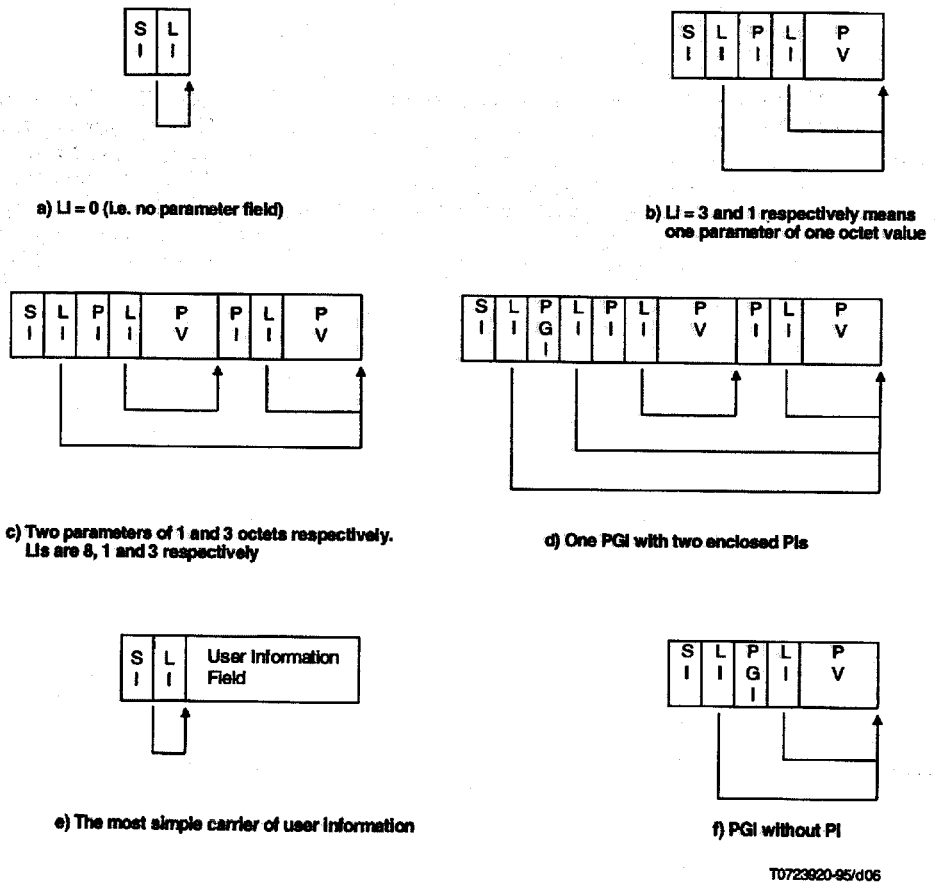

Figure 5 – Examples of SPDU structure

8.2.1 SPDUs

SPDUs shall contain, in the following order:

a) the SI field that identifies the type of SPDU (see Note);

b) the LI field that indicates the length of the associated parameter field defined in 8.2.1, c);

c) the parameter field which, if present, consists of the PGI units (see 8.2.2) and/or PI units (see 8.2.3) defined for the SPDU;

d) the user information field, if defined for the SPDU and if present.

NOTE – The SI field encompasses both the CI field and the RI field defined in Recommendation T.62. The protocol specified in this Recommendation | International Standard does not require a distinction to be made between these two fields.

8.2.2 PGI units

PGI units shall contain, in the following order:

a) the PGI field that identifies the parameter group;

b) the LI field that indicates the length of the associated parameter field defined in 8.2.2, c);

c) the parameter field which, if present, consists of either:

1) a single parameter value (see Note); or

2) one or more PI units (see 8.2.3).

NOTE – A PGI unit with one parameter is structurally equivalent to a PI unit, but the distinction has been retained in order to maintain compatibility with Recommendation T.62.

8.2.3 PI units

PI units shall contain, in the following order:

a) the PI field that identifies the parameter;

b) the LI field that indicates the length of the associated parameter field defined in 8.2.3, c);

c) the parameter field which, if present, consists of the parameter value.

8.2.4 Identifier fields

The SI field shall comprise one octet. The value of the SI field, specified as a decimal number in 8.3, shall be encoded as a binary number.

The PGI and PI fields shall each comprise one octet and shall contain a PGI or PI code respectively. The PGI and PI codes are expressed as decimal numbers in the tables in 8.3 and shall be encoded as a binary number.

8.2.5 Length indicator field

The value of the LI field is expressed as a binary number representing the length, in octets, of the associated parameter field (see Note). A value of zero indicates that the associated parameter field is absent.

LI fields indicating lengths within the range 0-254 shall comprise one octet.

LI fields indicating lengths within the range 255-65535 shall comprise three octets. The first octet shall be coded 1111 1111 and the second and third octets shall contain the length of the associated parameter field with the high order bits in the first of these two octets.

NOTE – The value of the LI field does not include either itself or any subsequent user information.

8.2.6 Parameter fields

PGI units and PI units defined as mandatory in the tables in 8.3 shall contain a parameter field of one or more octets.

Any PGI unit or PI unit defined as non-mandatory in the tables in 8.3 may be omitted if it is not required for conveying information (i.e. a parameter value). If a PGI unit or a PI unit contains an LI field with the value zero, the associated parameter field is absent (see Note) and the value of the parameter field shall be considered as its default value.

NOTE – It is recommended that if a non-mandatory parameter is absent, the associated PGI (or PI) and LI fields should not be included in the SPDU.

PGI units and PI units within the same nesting level shall be ordered in increasing value of their PGI and PI codes.

PGI or PI units containing:

a) a PGI or PI code listed in Annex B;

b) a PGI or PI code not listed in 8.3 or in Annex B,

are defined as valid.

NOTE – See A.4.3 for actions to be taken by the SPM on receipt of SPDUs containing these PGI or PI units.

8.2.7 Parameter values

Bits within a parameter field which are indicated as reserved shall have those bits set to zero in the SPDU.

NOTE – See A.4.3 for actions to be taken by the SPM on receipt of SPDUs containing such bits.

8.2.8 User information fields

Segments of a segmented SSDU shall be contained in the User Information Fields of SPDUs such that the order of the segments is maintained. An SSDU which is not segmented shall be contained in the User Information Field of a single SPDU. The order of the octets and the order of the bits in the SSDU shall be maintained in the SPDUs.

8.3 SPDU identifiers and associated parameter fields

The SPDUs specified in the remainder of this subclause do not, with certain exceptions, consider the case where an SSDU is segmented. When Protocol Version 2 is selected, most SSDUs may be segmented. (The circumstances under which an SSDU may be segmented are specified in 6.3.5). The additional encoding requirements when an SSDU is segmented are specified in 8.4.

8.3.1 CONNECT (CN) SPDU

8.3.1.1 The SI field shall contain the value 13.

8.3.1.2 The parameter fields shall be as specified in Table 11.

8.3.1.3 The Calling SS-user Reference PV field shall be as defined by the calling SS-user.

8.3.1.4 The Common Reference PV field shall be as defined by the calling SS-user.

8.3.1.5 The Additional Reference Information PV field shall be as defined by the calling SS-user.

8.3.1.6 If the Connect/Accept Item is absent, the default values defined for the enclosed PI units shall apply.

8.3.1.7 The Protocol Options PV field shall indicate whether or not the initiator is able to receive extended concatenated SPDUs (see 6.3.7). The encoding for this field shall be

a) bit 1 = 1: able to receive extended concatenated SPDUs;

b) bit 1 = 0: not able to receive extended concatenated SPDUs.

Bits 2-8 are reserved.

If the Protocol Options PI unit or PV field is absent, SPDUs with extended concatenation cannot be received.

8.3.1.8 The TSDU Maximum Size PV field shall be present if a TSDU Maximum Size is proposed. If the TSDU Maximum Size PV field is present:

a) the first two octets of the PV field shall contain the proposed maximum TSDU size, expressed in octets, in the direction from the initiator to the responder, encoded as a binary number, where the first of the two octets is the high order part of the number;

b) the second two octets of the PV field shall contain the proposed maximum TSDU size, expressed in octets, in the direction from the responder to the initiator, encoded as a binary number, where the first of the two octets is the high order part of the number.

If this parameter is not present, the TSDU Maximum Size is not limited over the session connection. If either pair of octets has the value zero, the TSDU size is not limited in the direction of transfer associated with that pair of octets.

Table 11 – Parameters of the CONNECT SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
Connection Identifier	nm	1	Calling SS-user Reference	nm	10	64 octets maximum	7.1.1 a) 1) 8.3.1.3
			Common Reference	nm	11	64 octets maximum	7.1.1 a) 2) 8.3.1.4
			Additional Reference Information	nm	12	4 octets maximum	7.1.1 a) 3) 8.3.1.5
Connect/Accept Item (see 8.3.1.6)	nm	5	Protocol Options	m	19	1 octet	7.1.1 b) 1) 8.3.1.7
			TSDU Maximum Size	nm	21	4 octets	7.1.1 b) 2) 8.3.1.8
			Version Number	m	22	1 octet	7.1.1 b) 3) 8.3.1.9
			Initial Serial Number	nm	23	6 octets maximum	7.1.1 b) 4) 8.3.1.10
			Token Setting Item	nm	26	1 octet	7.1.1 b) 5) 8.3.1.11
			Second Initial Serial Number	nm	55	6 octets maximum	7.1.1 b) 6) 8.3.1.12
			Upper Limit Serial Number	nm	56	(See reference)	7.1.1 b) 7) 8.3.1.13
			Large Initial Serial Number	nm	57	(See reference)	7.1.1 b) 8) 8.3.1.14
			Large Second Initial Serial Number	nm	58	(See reference)	7.1.1 b) 9) 8.3.1.15
			Session User Requirements	nm	20	2 octets	7.1.1 c) 8.3.1.16
			Calling Session Selector	nm	51	16 octets maximum	7.1.1 d) 8.3.1.17
			Called Session Selector	nm	52	16 octets maximum	7.1.1 d) 8.3.1.18
			Data Overflow	nm	60	1 octet	7.1.1 f) 8.3.1.19
User Data	nm	193				512 octets maximum	7.1.1 e) 1) 8.3.1.20
Extended User Data	nm	194				102 40 octets maximum	7.1.1 e) 2) 8.3.1.21

m Mandatory

nm Not mandatory (see 8.2.6)

8.3.1.9 The bits in the Version Number PV field shall indicate which protocol versions are proposed for use over this session connection:

a) bit 1: Protocol Version 1;

b) bit 2: Protocol Version 2.

Bits 3-8 are reserved.

The encoding for each bit shall be:

c) 0: use of protocol version not proposed;

d) 1: use of protocol version proposed.

If this PI unit or PV field is absent, the default shall be Protocol Version 1.

8.3.1.10 The Initial Serial Number PV field shall present if the activity management functional unit is not proposed and any of the minor synchronize, symmetric synchronize, major synchronize or resynchronize functional units are proposed. As an SS-user option, an Initial Serial Number PV field may be present if the activity management functional unit is proposed provided that any of the minor synchronize, symmetric synchronize, major synchronize or resynchronize functional units are also proposed. This parameter shall not be present if the Upper Limit Serial Number parameter is present.

Each digit of the serial number is encoded as an octet, as follows:

a) 0: 0011 0000;

b) 1: 0011 0001;

c) 2: 0011 0010;

d) 3: 0011 0011;

e) 4: 0011 0100;

f) 5: 0011 0101;

g) 6: 0011 0110;

h) 7: 0011 0111;

i) 8: 0011 1000;

j) 9: 0011 1001.

Serial number can range from 0 to (10**Upper Limit Serial Number) – 1. The most significant digit is encoded first in the PV field. Leading zeros may be omitted.

8.3.1.11 The Token Setting Item PV field, if present, shall indicate the initial position of the tokens. The bits of the Token Setting Item PV field are defined as bit pairs:

a) bits 8, 7: release token;

b) bits 6, 5: major/activity token;

c) bits 4, 3: synchronize-minor token;

d) bits 2, 1: data token.

The encoding for each bit pair shall be:

e) 00: initiator's side;

f) 01: responder's side;

g) 10: called SS user's choice;

h) 11: reserved.

The values are relevant only if the appropriate functional units are requested in the Session User Requirements parameter. If no functional unit requiring a token has been requested, this parameter need not be present.

If this PI unit or PV field is absent, the default shall be that all tokens whose availability is proposed in the Session User Requirements parameter are assigned to the calling SS-user.

8.3.1.12 Both the Initial Serial Number and Second Initial Serial Number PV fields may be present if symmetric synchronization is proposed. In this case the Initial Serial Number PV field indicates the First Initial Serial Number. This parameter shall not be present if the Upper Limit Serial Number parameter is present. The coding for these fields is defined in 8.3.1.10.

8.3.1.13 The Upper Limit Serial Number PV field, if present, shall indicate the maximum value of the serial number. Its maximum value is 6 and its default value is 6. A value of zero shall indicate that the maximum size of the serial number is unlimited. This PV field shall only be present under the conditions where an Initial Serial Number can be proposed (see 8.3.1.10). The length of this PV field shall be limited such that the total length (including SI and LI) of the SPDU does not exceed 65539 octets. This PV field shall be encoded as a binary number.

8.3.1.14 The Large Initial Serial Number PV field shall only be present under the conditions where an Initial Serial Number can be proposed (see 8.3.1.10) and if the Upper Limit Serial Number parameter is present. The length of this PV field shall be limited such that the total length (including SI and LI) of the SPDU does not exceed 65539 octets. The coding for this field is defined in 8.3.1.10.

8.3.1.15 The Large Second Initial Serial Number PV field shall only be present under the conditions where a Second Initial Serial Number can be proposed (see 8.3.1.12) and if the Upper Limit Serial Number parameter is present. The length of this PV field shall be limited such that the total length (including SI and LI) of the SPDU does not exceed 65539 octets. The coding for this field is defined in 8.3.1.10.

8.3.1.16 The bits in the Session User Requirements PV field shall indicate the functional units proposed by the calling SS-user, for use over this session connection:

a) bit 1: half-duplex functional unit;

b) bit 2: duplex functional unit;

c) bit 3: expedited data functional unit;

d) bit 4: minor synchronize functional unit;

e) bit 5: major synchronize functional unit;

f) bit 6: resynchronize functional unit;

g) bit 7: activity management functional unit;

h) bit 8: negotiated release functional unit;

i) bit 9: capability data exchange functional unit;

j) bit 10: exceptions functional unit;

k) bit 11: typed data functional unit;

l) bit 12: symmetric synchronize functional unit;

m) bit 13: data separation functional unit.

Bits 14-16 are reserved.

When this parameter is present, at least one of the half-duplex and the duplex functional units shall be proposed.

The encoding for each bit shall be

n) 0: use of functional unit not proposed;

o) 1: use of functional unit proposed.

When this parameter is absent, the default shall be as though bits 1, 4, 7, 9 and 10 are set to one and the remaining bits are set to zero.

8.3.1.17 The Calling Session Selector, if present, shall be derived from the Calling Session Address supplied by the calling SS-user.

8.3.1.18 The Called Session Selector, if present, shall be derived from the Called Session Address supplied by the calling SS-user.

8.3.1.19 The Data Overflow parameter, if present, shall indicate that there is more than 10 240 octets of user data to be transferred. This parameter shall not be present if Protocol Version 1 is proposed.

The encoding of this field shall be:

bit 1 = 1: more than 10 240 octets of user data

Bit 1 shall never be set equal to 0, Bits 2-8 are reserved.

If the Data Overflow PI unit or PV field is absent, there are not more than 10 240 octets of user data.

8.3.1.20 The User Data PV field, if present, shall contain user data supplied by the calling SS-user.

8.3.1.21 The Extended User Data parameter, if present, shall contain user data supplied by the calling SS-user. This parameter shall be present if the Data Overflow parameter is present. This parameter shall not be present if Protocol Version 1 is proposed.

Only one of the User Data and Extended User Data parameters may be present (see 7.1.1).

8.3.2 OVERFLOW ACCEPT (OA) SPDU

8.3.2.1 The SI field shall contain the value 16.

8.3.2.2 The parameter fields shall be as specified in Table 12.

Table 12 – Parameters of the OVERFLOW ACCEPT SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			TSDU Maximum Size	nm	21	4 octets	7.2.1 a) 8.3.2.3
			Version Number	m	22	1 octet	7.2.1 b) 8.3.2.4
m Mandatory nm Not mandatory (see 8.2.6)							

8.3.2.3 The TSDU Maximum Size parameter shall be present if a TSDU maximum size is proposed by the receiver. The encoding and default for this field is defined in 8.3.1.8.

8.3.2.4 In the Version Number PV field bit 2 shall have the value 1 indicating that Protocol Version 2 is proposed (and selected) for use over this session connection. Bit 1 shall have the value 0 indicating that Protocol Version 1 is not proposed.

Bits 3-8 are reserved.

8.3.3 CONNECT DATA OVERFLOW (CDO) SPDU

8.3.3.1 The SI field shall contain the value 15.

8.3.3.2 The parameter fields shall be as specified in Table 13.

Table 13 – Parameters of the CONNECT DATA OVERFLOW SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Enclosure Item	m	25	1 octet	7.3.1 a) 8.3.3.3
User Data	nm	193			22	65528 octets maximum	7.3.1 b) 8.3.3.4
m Mandatory nm Not mandatory (see 8.2.6)							

8.3.3.3 The Enclosure Item PV field shall indicate whether or not this SPDU is the end of the SSDU. The encoding for this field shall be:

a) bit 1 = 0: not beginning of SSDU;

b) bit 2 = 1: end of SSDU;
bit 2 = 0: not end of SSDU.

Bits 3-8 are reserved.

8.3.3.4 The User Data field, if present, shall contain a segment of the associated SSDU. The User Data field shall be present if the Enclosure Item has bit 2 = 0.

8.3.4 ACCEPT (AC) SPDU

8.3.4.1 The SI field shall contain the value 14.

8.3.4.2 The parameter fields shall be as specified in Table 14.

8.3.4.3 The Called SS-user Reference PV field shall be as defined by the called SS-user.

8.3.4.4 The Common Reference PV field shall be as defined by the called SS-user.

8.3.4.5 The Additional Reference Information PV field shall be as defined by the called SS-user.

8.3.4.6 If the Connect/Accept Item is absent, the default values defined for the enclosed PI units shall apply.

8.3.4.7 The Protocol Options PV field shall indicate whether or not the responder is able to receive extended concatenated SPDUs (see 6.3.7). The encoding and default for this field is defined in 8.3.1.7.

8.3.4.8 The TSDU Maximum Size parameter shall be present if a TSDU Maximum Size is proposed by the receiver. The encoding and default for this field is defined in 8.3.1.8. If an OVERFLOW ACCEPT SPDU has been sent previously, the TSDU Maximum Size parameter shall have the same value as was indicated in the OVERFLOW ACCEPT SPDU.

8.3.4.9 The Version Number PV field shall have the value and encoding specified in 8.3.1.9. If an OVERFLOW ACCEPT SPDU has been sent previously on this session connection, then the Version Number parameter shall have the same value as was indicated in the OVERFLOW ACCEPT SPDU.

8.3.4.10 The Initial Serial Number PV field shall only be present if the activity management functional unit is not selected and if any of the following functional units are selected:

a) minor synchronize functional unit;

b) symmetric synchronize functional unit;

c) major synchronize functional unit;

d) resynchronize functional unit.

This parameter shall not be present if the Upper Limit Serial Number parameter is present.

The encoding for the Initial Serial Number PV field is defined in 8.3.1.10.

8.3.4.11 The Token Setting Item PV field indicates the initial token settings for each token available on this session connection. The bits and encoding are defined in 8.3.1.11. In the case where the initial assignment of the related token was indicated as the called SS-user's choice (in the Token Setting Item PV field of the associated CONNECT SPDU), the field shall contain the value chosen by the called SS-user. Otherwise, the values set in the CONNECT SPDU shall be returned. The value "called SS-user's choice" is not a permitted value in the ACCEPT SPDU. The values are relevant only if the appropriate functional units are requested in the Session User Requirements parameter. If no functional unit requiring a token has been requested, this parameter need not be present.

8.3.4.12 Both the Initial Serial Number and Second Initial Serial Number PV fields shall be present if symmetric synchronization is selected and the activity management functional unit is not selected. In this case the Initial Serial Number PV field indicates the First Initial Serial Number. This parameter shall not be present if the Upper Limit Serial Number parameter is present. The coding for these fields is defined in 8.3.1.10.

Table 14 – Parameters of the ACCEPT SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
Connection Identifier	nm	1	Called SS-user Reference	nm	9	64 octets maximum	7.4.1 a) 1) 8.3.4.3
			Common Reference	nm	11	64 octets maximum	7.4.1 a) 2) 8.3.4.4
			Additional Reference Information	nm	12	4 octets maximum	7.4.1 a) 3) 8.3.4.5
Connect/Accept Item (see 8.3.4.6)	nm	5	Protocol Options	m	19	1 octet	7.4.1 b) 1) 8.3.4.7
			TSDU Maximum Size	nm	21	4 octets	7.4.1 b) 2) 8.3.4.8
			Version Number	m	22	1 octet	7.4.1 b) 3) 8.3.4.9
			Initial Serial Number	nm	23	6 octets maximum	7.4.1 b) 4) 8.3.4.10
			Token Setting Item	nm	26	1 octet	7.4.1 b) 5) 8.3.4.11
			Second Initial Serial Number	nm	55	6 octets maximum	7.4.1 b) 6) 8.3.4.12
			Upper Limit Serial Number	nm	56	(See reference)	7.4.1 b) 7) 8.3.4.13
			Large Initial Serial Number	nm	57	(See reference)	7.4.1 b) 8) 8.3.4.14
			Large Second Initial Serial Number	nm	58	(See reference)	7.4.1 b) 9) 8.3.4.15
			Token Item	nm	16	1 octet	7.4.1 c) 8.3.4.16
			Session User Requirements	nm	20	2 octets	7.4.1 d) 8.3.4.17
			Enclosure Item	nm	25	1 octet	7.4.1 e) 8.3.4.18
			Calling Session Selector	nm	51	16 octets maximum	7.4.1 f) 8.3.4.19
			Responding Session Selector	nm	52	16 octets maximum	7.4.1 f) 8.3.4.20
User Data	nm	193				(See reference)	7.4.1 g) 8.3.4.21

m Mandatory
nm Not mandatory (see 8.2.6)

8.3.4.13 The Upper Limit Serial Number PV field, if present, shall indicate the maximum value of the serial number. Its maximum value is 6 and its default value is 6. A value of zero shall indicate that the maximum size of the serial number is unlimited. This PV field shall only be present under the conditions where an Initial Serial Number can be present (see 8.3.4.10). The length of this PV field shall be limited such that the total length (including SI and LI) of the SPDU does not exceed 654539 octets. This PV field shall be encoded as a binary number.

8.3.4.14 The Large Initial Serial Number PV field shall only be present under the conditions where an Initial Serial Number can be present (see 8.3.4.10) and if the Upper Limit Serial Number parameter is present. The length of this PV field shall be limited such that the total length (including SI and LI) of the SPDU does not exceed 65539 octets. The coding for this field is defined in 8.3.1.10.

8.3.4.15 The Large Second Initial Serial Number PV field shall only be present under the conditions where a Second Initial Serial Number can be present (see 8.3.4.12) and if the Upper Limit Serial Number parameter is present. The length of this PV field shall be limited such that the total length (including SI and LI) of the SPDU does not exceed 65539 octets. The coding for this field is defined in 8.3.1.10.

8.3.4.16 The Token Item PV field, if present, shall indicate which tokens are requested by the called SS-user:

a) bit 7 = 1: release token;

b) bit 5 = 1: major/activity token;

c) bit 3 = 1: synchronize minor token;

d) bit 1 = 1: data token.

Bits 2, 4, 6 and 8 are reserved.

Bits corresponding to tokens which are not available are ignored.

8.3.4.17 The bits in the Session User Requirements PV field shall indicate the functional units proposed by the called SS-user, for use over this session connection. This PV field shall not have both bit 1 set (half-duplex functional unit) and bit 2 set (duplex functional unit), but the chosen bit shall have been set in the CONNECT SPDU. The Session User Requirements PV field shall not have both bit 4 set (minor synchronize functional unit) and bit 12 set (symmetric synchronize functional unit) in the ACCEPT SPDU. At most one of these two bits can be set in the ACCEPT SPDU. The encoding and default value are defined in 8.3.1.16.

8.3.4.18 The Enclosure Item parameter, if present, shall indicate that the SPDU is the beginning, but not end of the SSDU. This parameter shall not be present if Protocol Version 1 is selected. The encoding for this field shall be:

a) bit 1 = 1: beginning of SSDU;

b) bit 2 = 0: not end of SSDU.

Bits 3-8 are reserved.

See 8.4.2 for encoding subsequent SPDUs in the sequence.

8.3.4.19 The Calling Session Selector, if present, shall be the same value as in the CONNECT SPDU. When this parameter is absent, the default shall be as though this parameter was set to the value of the Calling Session Selector in the CONNECT SPDU.

8.3.4.20 The Responding Session Selector, if present, shall be derived from the Responding Session Address supplied by the called SS-user. When this parameter is absent, the default shall be as though this parameter was set to a null value.

8.3.4.21 The User Data PV field, if present, shall contain user data supplied by the called SS-user. The length of the User Data parameter shall not exceed 512 octets if Protocol Version 1 has been selected, and shall be limited such that the total length (including SI and LI) of the SPDU does not exceed 65539 octets if Protocol Version 2 is selected. If the Enclosure Item parameter is present, the User Data parameter is mandatory.

8.3.5 REFUSE (RF) SPDU

8.3.5.1 The SI field shall contain the value 12.

8.3.5.2 The parameter fields shall be as specified in Table 15.

Table 15 – Parameters of the REFUSE SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
Connection Identifier	nm	1	Called SS-user Reference	nm	9	64 octets maximum	7.5.1 a) 1) 8.3.5.3
			Common Reference	nm	11	64 octets maximum	7.5.1 a) 2) 8.3.5.4
			Additional Reference Information	nm	12	4 octets maximum	7.5.1 a) 3) 8.3.5.5
			Transport Disconnect	nm	17	1 octet	7.5.1 b) 8.3.5.6
			Session User Requirements	nm	20	2 octets	7.5.1 c) 8.3.5.7
			Version Number	nm	22	1 octet	7.5.1 d) 8.3.5.8
			Enclosure Item	nm	25	1 octet	7.5.1 e) 8.3.5.9
			Reason Code	nm	50	(See reference)	7.5.1 f) 8.3.5.10

m Mandatory
nm Not mandatory (see 8.2.6)

8.3.5.3 The Called SS-user Reference PV field shall be as defined by the called SS-user.

8.3.5.4 The Common Reference PV field shall be as defined by the called SS-user.

8.3.5.5 The Additional Reference Information PV field shall be as defined by the called SS-user.

8.3.5.6 The Transport Disconnect PV field shall indicate whether or not the transport connection is to be kept. The encoding for this field shall be:

a) bit 1 = 0: transport connection is kept;

b) bit 1 = 1: transport connection is released.

Bits 2-8 are reserved.

If this parameter is absent, the transport connection is released.

8.3.5.7 The Session User Requirements PV field shall only be present if the Reason Code is 2 and shall indicate the functional units required by the called SS-user and supported by the responder. The encoding shall be the same as in the CONNECT SPDU (see 8.3.1.16).

8.3.5.8 The Version Number PV field shall have the value and encoding specified in 8.3.1.9. If an OVERFLOW ACCEPT SPDU has been sent previously on this session connection, then the Version Number parameter shall have the same value as was indicated in the OVERFLOW ACCEPT SPDU.

8.3.5.9 The Enclosure Item parameter, if present, shall be encoded as specified in 8.3.4.18. This parameter shall not be present if Protocol Version 1 is selected.

8.3.5.10 The Reason Code PV field shall contain a reason code in the first octet. Depending on the value of this first octet, additional octets may be used. The following values are defined for the first octet:

a) 0: Rejection by called SS-user; reason not specified.

b) 1: Rejection by called SS-user due to temporary congestion.

c) 2: Rejection by called SS-user. Subsequent octets may be used for user data up to a length of 512 octets if Protocol Version 1 has been selected, and up to a length such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets if Protocol Version 2 has been selected.

d) * 128 + 1: Session Selector unknown.

e) * 128 + 2: SS-user not attached to SSAP.

f) 128 + 3: SPM congestion at connect time.

g) * 128 + 4: Proposed protocol versions not supported.

h) * 128 + 5: Rejection by the SPM; reason not specified.

i) * 128 + 6: Rejection by the SPM; implementation restriction stated in the PICS.

NOTE – Reasons marked with an asterisk (*) may be reported to the SS-user as persistent, others reported as transient.

All other values are reserved.

The Session User Requirements parameter may only be present if the value of the Reason Code is 2. If the Reason Code has the value 2 and the Session User Requirements parameter is not present, the default value shall be assumed (see 8.3.1.16).

If the Enclosure Item parameter is present, the Reason Code parameter is mandatory and shall be followed by octets of user data.

8.3.6 FINISH (FN) SPDU

8.3.6.1 The SI field shall contain the value 9.

8.3.6.2 The parameter fields shall be as specified in Table 16.

Table 16 – Parameters of the FINISH SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Transport Disconnect	nm	17	1 octet	7.6.1 a) 8.3.6.3
			Enclosure Item	nm	25	1 octet	7.6.1 b) 8.3.6.4
User Data	nm	193				(See reference)	7.6.1 c) 8.3.6.5
m Mandatory nm Not mandatory (see 8.2.6)							

8.3.6.3 The Transport Disconnect PV field shall indicate whether or not the transport connection is to be kept. The encoding for this field shall be:

a) bit 1 = 0: transport connection is kept;

b) bit 1 = 1: transport connection is released.

Bits 2-8 are reserved.

If this parameter is absent, the transport connection shall be released.

8.3.6.4 The Enclosure Item parameter, if present, shall be encoded as specified in 8.3.4.18. This parameter shall not be present if Protocol Version 1 is selected.

8.3.6.5 The User Data PV field, if present, shall contain user data supplied by the called SS-user. The length of the User Data parameter shall not exceed 512 octets if Protocol Version 1 has been selected, and shall be limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets if Protocol Version 2 is selected. If the Enclosure Item parameter is present, the User Data parameter is mandatory.

8.3.7 DISCONNECT (DN) SPDU

8.3.7.1 The SI field shall contain the value 10.

8.3.7.2 The parameter field shall be as specified in Table 17.

Table 17 – Parameters of the DISCONNECT SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Enclosure Item	nm	25	1 octet	7.7.1 a) 8.3.7.3
User Data	nm	193				(See reference)	7.7.1 b) 8.3.7.4
m Mandatory nm Not mandatory (see 8.2.6)							

8.3.7.3 The Enclosure Item parameter, if present, shall be encoded as specified in 8.3.4.18. This parameter shall not be present if Protocol Version 1 is selected.

8.3.7.4 The User Data PV field, if present, shall contain user data supplied by the called SS-user. The length of the User Data parameter shall not exceed 512 octets if Protocol Version 1 has been selected, and shall be limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets if Protocol Version 2 is selected. If the Enclosure Item parameter is present, the User Data parameter is mandatory.

8.3.8 NOT FINISHED (NF) SPDU

8.3.8.1 The SI field shall contain the value 8.

8.3.8.2 The parameter field shall be as specified in Table 18.

Table 18 – Parameters of the NOT FINISHED SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Enclosure Item	nm	25	1 octet	7.8.1 a) 8.3.8.3
User Data	nm	193				(See reference)	7.8.1 b) 8.3.8.4
m Mandatory nm Not mandatory (see 8.2.6)							

8.3.8.3 The Enclosure Item parameter, if present, shall be encoded as specified in 8.3.4.18. This parameter shall not be present if Protocol Version 1 is selected.

8.3.8.4 The User Data PV field, if present, shall contain user data supplied by the called SS-user. The length of the User Data parameter shall not exceed 512 octets if Protocol Version 1 has been selected, and shall be limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets if Protocol Version 2 is selected. If the Enclosure Item parameter is present, the User Data parameter is mandatory.

8.3.9 ABORT (AB) SPDU

8.3.9.1 The SI field shall contain the value 25.

8.3.9.2 The parameter fields shall be as specified in Table 19.

Table 19 – Parameters of the ABORT SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Transport Disconnect	m	17	1 octet	7.9.1.1 a) 7.9.1.2 a) 8.3.9.3
			Enclosure Item	nm	25	1 octet	7.9.1.2 b) 7.9.1.2 d) 8.3.9.4
			Reflect Parameter Values	nm	49	9 octets maximum	7.9.1.1 b) 8.3.9.5
User Data	nm	193				(See reference)	7.9.1.1 c) 7.9.1.2 c) 7.9.1.2 e) 8.3.9.6
m Mandatory nm Not mandatory (see 8.2.6)							

8.3.9.3 The Transport Disconnect PV field shall indicate whether or not the transport connection is to be kept, together with an optional reason code. The encoding for this field shall be:

a) bit 1 = 0: transport connection is kept;

b) bit 1 = 1: transport connection is released;

c) bit 2 = 1: user abort (see 8.3.9.6);

d) bit 3 = 1: protocol error (see 8.3.9.5);

e) bit 4 = 1: no reason;

f) bit 5 = 1: implementation restriction stated in the PICS.

Bits 6-8 are reserved.

8.3.9.4 The Enclosure Item parameter, if present, shall be encoded as specified in 8.3.4.18. This parameter shall not be present if Protocol Version 1 is selected.

8.3.9.5 The Reflect Parameter Values PV field shall only be present if the Transport Disconnect PV field indicates protocol error and shall contain an implementation defined value and semantics.

8.3.9.6 The User Data PV field shall only be present if the Transport Disconnect PV field indicates user abort and shall contain user data supplied by the SS-user. If this SPDU is to be sent on the transport expedited flow, the length of the User Data parameter is limited to 9 octets and the Enclosure Item shall not be present. If the SPDU is to be sent on the transport normal flow, the length of the User Data parameter is limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets.

8.3.10 ABORT ACCEPT (AA) SPDU

8.3.10.1 The SI field shall contain the value 26.

8.3.10.2 There is no parameter field associated with this SPDU.

8.3.11 DATA TRANSFER (DT) SPDU

8.3.11.1 The SI field shall contain the value 1.

8.3.11.2 The parameter field shall be as specified in Table 20.

Table 20 – Parameters of the DATA TRANSFER SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Enclosure Item	nm	25	1 octet	7.11.1 a) 8.3.11.3
User Information Field						Unlimited	7.11.1 b) 8.3.11.4
m Mandatory nm Not mandatory (see 8.2.6)							

8.3.11.3 The Enclosure Item PV field, if present, shall indicate whether or not this SPDU is the beginning or end of the SSDU. This field shall be present if segmenting has been selected. This field shall not be present if segmenting has not been selected. The encoding for this field shall be:

a) bit 1 = 1: beginning of SSDU;
   bit 1 = 0: not beginning of SSDU;

b) bit 2 = 1: end of SSDU;
   bit 2 = 0: not end of SSDU.

Bits 3-8 are reserved.

If this field is not present, segmenting has not been selected and this SPDU contains a complete SSDU.

8.3.11.4 The User Information Field, if present, shall contain user data supplied by the SS-user. The User Information Field shall be present if the Enclosure Item is not present, or has bit 2 = 0.

8.3.12 EXPEDITED (EX) SPDU

8.3.12.1 The SI field shall contain the value 5.

8.3.12.2 This SPDU contains only a User Information Field as specified in Table 21.

8.3.12.3 The User Information Field shall contain user data supplied by the SS-user.

Table 21 – Parameters of the EXPEDITED SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
User Information Field						14 octets maximum	7.12.1 8.3.12.3
m Mandatory nm Not mandatory (see 8.2.6)							

8.3.13 TYPED DATA (TD) SPDU

8.3.13.1 The SI field shall contain the value 33.

8.3.13.2 The parameter field shall be as specified in Table 22.

Table 22 – Parameters of the TYPED DATA SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Enclosure Item	nm	25	1 octet	7.13.1 a) 8.3.13.3
User Information Field						Unlimited	7.13.1 b) 8.3.13.4
m Mandatory nm Not mandatory (see 8.2.6)							

8.3.13.3 The Enclosure Item PV field, if present, shall indicate whether or not this SPDU is the beginning or end of the SSDU. This field shall be present if segmenting has been selected. This field shall not be present if segmenting has not been selected. The encoding for this field shall be:

a) bit 1 = 1: beginning of SSDU;
   bit 1 = 0: not beginning of SSDU;

b) bit 2 = 1: end of SSDU;
   bit 2 = 0: not end of SSDU.

Bits 3-8 are reserved.

If this field is not present, segmenting has not been selected and this SPDU contains a complete SSDU.

8.3.13.4 The User Information Field, if present, shall contain user data supplied by the SS-user. The User Information Field shall be present if the Enclosure Item is not present, or has bit 2 = 0.

8.3.14 CAPABILITY DATA (CD) SPDU

8.3.14.1 The SI field shall contain the value 61.

8.3.14.2 The parameter field shall be as specified in Table 23.

Table 23 – Parameters of the CAPABILITY DATA SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Enclosure Item	nm	25	1 octet	7.14.1 a) 8.3.14.3
User Data	nm	193				(See reference)	7.14.1 b) 8.3.14.4
m Mandatory nm Not mandatory (see 8.2.6)							

8.3.14.3 The Enclosure Item parameter, if present, shall be encoded as specified in 8.3.4.18. This parameter shall not be present if Protocol Version 1 is selected.

8.3.14.4 The User Data PV field, if present, shall contain user data supplied by the called SS-user. The length of the User Data parameter shall not exceed 512 octets if Protocol Version 1 has been selected, and shall be limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets if Protocol Version 2 is selected. If the Enclosure Item parameter is present, the User Data parameter is mandatory.

8.3.15 CAPABILITY DATA ACK (CDA) SPDU

8.3.15.1 The SI field shall contain the value 62.

8.3.15.2 The parameter field shall be as specified in Table 24.

Table 24 – Parameters of the CAPABILITY DATA ACK SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Enclosure Item	nm	25	1 octet	7.15.1 a) 8.3.15.3
User Data	nm	193				(See reference)	7.15.1 b) 8.3.15.4
m Mandatory nm Not mandatory (see 8.2.6)							

8.3.15.3 The Enclosure Item parameter, if present, shall be encoded as specified in 8.3.4.18. This parameter shall not be present if Protocol Version 1 is selected.

8.3.15.4 The User Data PV field, if present, shall contain user data supplied by the called SS-user. The length of the User Data parameter shall not exceed 512 octets if Protocol Version 1 has been selected, and shall be limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets if Protocol Version 2 is selected. If the Enclosure Item parameter is present, the User Data parameter is mandatory.

8.3.16 GIVE TOKENS (GT) SPDU

8.3.16.1 The SI field shall contain the value 1.

8.3.16.2 The parameter field shall be as specified in Table 25.

Table 25 – Parameters of the GIVE TOKENS SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Token Item	nm	16	1 octet	7.16.1 a) 8.3.16.3
			Enclosure Item	nm	25	1 octet	7.16.1 b) 8.3.16.4
User Data	nm	193				(See reference)	7.16.1 c) 8.3.16.5
m Mandatory nm Not mandatory (see 8.2.6)							

8.3.16.3 The Token Item PV field, if present, shall indicate which tokens are being given by the sending SS-user:

a) bit 7 = 1: release token;

b) bit 5 = 1: major/activity token;

c) bit 3 = 1: synchronize minor token;

d) bit 1 = 1: data token.

Bits 2, 4, 6 and 8 are reserved.

Bits corresponding to tokens which are not available shall be zero.

If this PV field is present, at least one bit corresponding to an available token shall be set to one.

8.3.16.4 The Enclosure Item parameter, if present, shall be encoded as specified in 8.3.4.18. This parameter shall not be present if Protocol Version 1 is selected.

8.3.16.5 The User Data PV field, if present, shall contain user data supplied by the SS-user. This PGI unit shall only be present if the Token Item PI unit is present and shall not be present if Protocol Version 1 is selected. The length of the User Data parameter is limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets.

8.3.16.6 This SPDU may be used without the Token Item PI unit when concatenated with Category 2 SPDUs according to Tables 7 and 8. With some concatenations (see Tables 7 and 8) the Token Item PI unit must be absent.

8.3.17 PLEASE TOKENS (PT) SPDU

8.3.17.1 The SI field shall contain the value 2.

8.3.17.2 The parameter fields shall be as specified in Table 26.

Table 26 – Parameters of the PLEASE TOKENS SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Token Item	nm	16	1 octet	7.17.1 a) 8.3.17.3
			Enclosure Item	nm	25	1 octet	7.17.1 b) 8.3.17.4
User Data	nm	193				(See reference)	7.17.1 c) 8.3.17.5
m Mandatory nm Not mandatory (see 8.2.6)							

8.3.17.3 The Token Item PV field, if present, shall indicate which tokens are being requested by the sending SS-user:

a) bit 7 = 1: release token;

b) bit 5 = 1: major/activity token;

c) bit 3 = 1: synchronize minor token;

d) bit 1 = 1: data token.

Bits 2, 4, 6 and 8 are reserved.

Bits corresponding to tokens which are not available shall be zero.

If this PV field is present, at least one bit corresponding to an available token shall be set to one.

8.3.17.4 The Enclosure Item parameter, if present, shall be encoded as specified in 8.3.4.18. This parameter shall not be present if Protocol Version 1 is selected.

8.3.17.5 The User Data PV field, if present, shall contain user data supplied by the called SS-user. This PGI unit shall only be present if the Token Item PI unit is present. The length of the User Data parameter shall not exceed 512 octets if Protocol Version 1 has been selected, and shall be limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets if Protocol Version 2 is selected. If the Enclosure Item parameter is present, the User Data parameter is mandatory.

8.3.17.6 This SPDU may be used without the Token Item PI unit and the User Data PGI unit when concatenated with Category 2 SPDUs according to Tables 7 and 8. In this case, the SPDU does not achieve any Please Token function.

8.3.18 GIVE TOKENS CONFIRM (GTC) SPDU

8.3.18.1 The SI field shall contain the value 21.

8.3.18.2 The parameter fields shall be as specified in Table 27.

Table 27 – Parameters of the GIVE TOKENS CONFIRM SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Enclosure Item	nm	25	1 octet	7.18.1 a) 8.3.18.3
User Data	nm	193				(See reference)	7.18.1 b) 8.3.18.4
m Mandatory nm Not mandatory (see 8.2.6)							

8.3.18.3 The Enclosure Item parameter, if present, shall be encoded as specified in 8.3.4.18. This parameter shall not be present if Protocol Version 1 is selected.

8.3.18.4 The User Data PV field, if present, shall contain user data supplied by the SS-user. This PGI unit shall not be present if Protocol Version 1 is selected. The length of the User Data parameter is limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets.

8.3.19 GIVE TOKENS ACK (GTA) SPDU

8.3.19.1 The SI field shall contain the value 22.

8.3.19.2 There is no parameter field associated with this SPDU.

8.3.20 MINOR SYNC POINT (MIP) SPDU

8.3.20.1 The SI field shall contain the value 49.

8.3.20.2 The parameter fields shall be as specified in Table 28.

Table 28 – Parameters of the MINOR SYNC POINT SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Sync Type Item	nm	15	1 octet	7.20.1 a) 8.3.20.3
			Enclosure Item	nm	25	1 octet	7.20.1 b) 8.3.20.4
			Serial Number	m	42	(See reference)	7.20.1 c) 8.3.20.5
User Data	nm	193				(See reference)	7.20.1 d) 8.3.20.6
m Mandatory nm Not mandatory (see 8.2.6)							

8.3.20.3 The Sync Type Item PV field, if present, shall indicate the following:

a) bit 1 = 1: explicit confirmation not required;
   bit 1 = 0: explicit confirmation required;

b) bit 2 = 1: data separation required;
   bit 2 = 0: data separation not required.

Bits 3-8 are reserved.

This parameter field shall be absent if an explicit confirmation is required and data separation is not required.

8.3.20.4 The Enclosure Item parameter, if present, shall be encoded as specified in 8.3.4.18. This parameter shall not be present if Protocol Version 1 is selected.

8.3.20.5 The Serial Number PV field shall be coded as specified in 8.3.1.10. The length of this PV field shall not exceed the length specified by the Upper Limit Serial Number (see 8.3.4.13) and shall be limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets.

8.3.20.6 The User Data PV field, if present, shall contain user data supplied by the called SS-user. The length of the User Data parameter shall not exceed 512 octets if Protocol Version 1 has been selected, and shall be limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets if Protocol Version 2 is selected. If the Enclosure Item parameter is present, the User Data parameter is mandatory.

8.3.21 MINOR SYNC ACK (MIA) SPDU

8.3.21.1 The SI field shall contain the value 50.

8.3.21.2 The parameter fields shall be as specified in Table 29.

Table 29 – Parameters of the MINOR SYNC ACK SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Enclosure Item	nm	25	1 octet	7.21.1 a) 8.3.21.3
			Serial Number	m	42	(See reference)	7.21.1 b) 8.3.21.4
			User Data	nm	46	(See reference)	7.21.1 c) 8.3.21.5
m Mandatory nm Not mandatory (see 8.2.6)							

8.3.21.3 The Enclosure Item parameter, if present, shall be encoded as specified in 8.3.4.18. This parameter shall not be present if Protocol Version 1 is selected.

8.3.21.4 The Serial Number PV field shall be coded as specified in 8.3.1.10. The length of this PV field shall not exceed the length specified by the Upper Limit Serial Number (see 8.3.4.13) and shall be limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets.

8.3.21.5 The User Data PV field, if present, shall contain user data supplied by the called SS-user. The length of the User Data parameter shall not exceed 512 octets if Protocol Version 1 has been selected, and shall be limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets if Protocol Version 2 is selected. If the Enclosure Item parameter is present, the User Data parameter is mandatory.

8.3.22 MAJOR SYNC POINT (MAP) SPDU

8.3.22.1 The SI field shall contain the value 41. This is the same value as the SI field for the ACTIVITY END SPDU (see 8.3.35).

8.3.22.2 The parameter fields shall be as specified in Table 30.

Table 30 – Parameters of the MAJOR SYNC POINT SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Sync Type Item	m	15	1 octet	7.22.1 a) 8.3.22.3
			Enclosure Item	nm	25	1 octet	7.22.1 b) 8.3.22.4
			Serial Number	m	42	(See reference)	7.22.1 c) 8.3.22.5
User Data	nm	193				(See reference)	7.22.1 d) 8.3.22.6
m Mandatory nm Not mandatory (see 8.2.6)							

8.3.22.3 The Sync Type Item PV field shall indicate that this is not the end of an activity:

bit 1 = 1: major synchronization point without end of activity.

Bits 2-8 are reserved.

8.3.22.4 The Enclosure Item parameter, if present, shall be encoded as specified in 8.3.4.18. This parameter shall not be present if Protocol Version 1 is selected.

8.3.22.5 The Serial Number PV field shall be coded as specified in 8.3.1.10. The length of this PV field shall not exceed the length specified by the Upper Limit Serial Number (see 8.3.4.13) and shall be limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets.

8.3.22.6 The User Data PV field, if present, shall contain user data supplied by the called SS-user. The length of the User Data parameter shall not exceed 512 octets if Protocol Version 1 has been selected, and shall be limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets if Protocol Version 2 is selected. If the Enclosure Item parameter is present, the User Data parameter is mandatory.

8.3.23 MAJOR SYNC ACK (MAA) SPDU

8.3.23.1 The SI field shall contain the value 42.

8.3.23.2 The parameter fields shall be as specified in Table 31.

Table 31 – Parameters of the MAJOR SYNC ACK SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Enclosure Item	nm	25	1 octet	7.23.1 a) 8.3.23.3
			Serial Number	nm	42	(See reference)	7.23.1 b) 8.3.23.4
			Second Serial Number	nm	54	(See reference)	7.23.1 c) 8.3.23.5
User Data	nm	193				(See reference)	7.23.1 d) 8.3.23.6
m Mandatory nm Not mandatory (see 8.2.6)							

8.3.23.3 The Enclosure Item parameter, if present, shall be encoded as specified in 8.3.4.18. This parameter shall not be present if Protocol Version 1 is selected.

8.3.23.4 The Serial Number PV field shall be coded as specified in 8.3.1.10. The length of this PV field shall not exceed the length specified by the Upper Limit Serial Number (see 8.3.4.13) and shall be limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets. This PV field shall be present if and only if the symmetric synchronize functional unit has not been selected.

8.3.23.5 The Second Serial number PV field shall be coded as specified in 8.3.1.10. The length of this PV field shall not exceed the length specified by the Upper Limit Serial Number (see 8.3.4.13) and shall be limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets. This PV field shall be present if and only if the symmetric synchronize functional unit has been selected.

8.3.23.6 The User Data PV field, if present, shall contain user data supplied by the called SS-user. The length of the User Data parameter shall not exceed 512 octets if Protocol Version 1 has been selected, and shall be limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets if Protocol Version 2 is selected. If the Enclosure Item parameter is present, the User Data parameter is mandatory.

NOTE – This SPDU is identical to the ACTIVITY END ACK SPDU (see 8.3.36).

8.3.24 RESYNCHRONIZE (RS) SPDU

8.3.24.1 The SI field shall contain the value 53.

8.3.24.2 The parameter fields shall be as specified in Table 32.

Table 32 – Parameters of the RESYNCHRONIZE SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Enclosure Item	nm	25	1 octet	7.24.1 a) 8.3.24.3
			Token Setting Item	nm	26	1 octet	7.24.1 b) 8.3.24.4
			Resync Type	nm	27	1 octet	7.24.1 c) 8.3.24.5
			Serial Number	nm	42	(See reference)	7.24.1 d) 8.3.24.6
			Second Resync Type	nm	53	1 octet	7.24.1 e) 8.3.24.7
			Second Serial Number	nm	54	(See reference)	7.24.1 f) 8.3.24.8
User Data	nm	193				(See reference)	7.24.1 g) 8.3.24.9

m Mandatory
nm Not mandatory (see 8.2.6)

8.3.24.3 The Enclosure Item parameter, if present, shall be encoded as specified in 8.3.4.18. This parameter shall not be present if Protocol Version 1 is selected.

8.3.24.4 The Token Setting Item PV field indicates the requesting SS-user's proposed settings for each available token. The bits of the Token Setting Item PV field are defined as bit pairs:

a) bits 8, 7: release token;

b) bits 6, 5: major/activity token;

c) bits 4, 3: synchronize-minor token;

d) bits 2, 1: data token.

The encoding for each bit pair shall be:

e) 00: requestor's side;

f) 01: acceptor's side;

g) 10: accepting SS-user's choice;

h) 11: reserved.

The values are relevant only if the appropriate token is available. If no token is available, this parameter need not be present.

8.3.24.5 The Resync Type PV field indicates the resynchronize type which is required:

a) 0: resynchronize restart;

b) 1: resynchronize abandon;

c) 2: resynchronize set.

All other values are reserved.

This PV field shall be present if the symmetric synchronize functional unit is not selected. This PV field may be present if the symmetric synchronize functional unit is selected.

8.3.24.6 The Serial Number PV field shall be encoded as specified in 8.3.1.10. The length of this PV field shall not exceed the length specified by the Upper Limit Serial Number (see 8.3.4.13) and shall be limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets. This PV field shall be present if the symmetric synchronize functional unit is not selected. This PV field may be present if the symmetric synchronize functional unit is selected.

8.3.24.7 The Second Resync Type PV field shall be coded as specified in 8.3.24.5. This PV field may only be present if the symmetric synchronize functional unit is selected.

8.3.24.8 The Second Serial Number PV field shall be coded as specified in 8.3.1.10. The length of this PV field shall not exceed the length specified by the Upper Limit Serial Number (see 8.3.4.13) and shall be limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets. This PV field may only be present if the symmetric synchronize functional unit is selected.

8.3.24.9 The User Data PV field, if present, shall contain user data supplied by the called SS-user. The length of the User Data parameter shall not exceed 512 octets if Protocol Version 1 has been selected, and shall be limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets if Protocol Version 2 is selected. If the Enclosure Item parameter is present, the User Data parameter is mandatory.

8.3.25 RESYNCHRONIZE ACK (RA) SPDU

8.3.25.1 The SI field shall contain the value 34.

8.3.25.2 The parameter fields shall be as specified in Table 33.

8.3.25.3 The Enclosure Item parameter, if present, shall be encoded as specified in 8.3.4.18. This parameter shall not be present if Protocol Version 1 is selected.

8.3.25.4 The Token Setting Item PV field indicates token settings for each token available on the session connection. The bits and encoding are defined in 8.3.24.4. For the case where the requesting SS-user has indicated that the assignment is the accepting SS-user's choice, the field shall contain the values chosen by the accepting SS-user. Otherwise, the values in the RESYNCHRONIZE SPDU shall be returned.

This parameter need not be present if no tokens are available.

8.3.25.5 The First Resync Type PV field shall be coded as specified in 8.3.24.5. This PV field may only be present if the symmetric synchronize functional unit is selected.

8.3.25.6 The Serial Number PV field shall be coded as specified in 8.3.1.10. The length of this PV field shall not exceed the length specified by the Upper Limit Serial Number (see 8.3.4.13) and shall be limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets. This PV field shall be present if the symmetric synchronize functional unit is not selected. This PV field may be present if the symmetric synchronize functional unit is selected.

Table 33 – Parameters of the RESYNCHRONIZE ACK SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Enclosure Item	nm	25	1 octet	7.25.1 a) 8.3.25.3
			Token Setting Item	nm	26	1 octet	7.25.1 b) 8.3.25.4
			Resync Type	nm	27	1 octet	7.25.1 c) 8.3.25.5
			Serial Number	nm	42	(See reference)	7.25.1 d) 8.3.25.6
			Second Resync Type	nm	53	1 octet	7.25.1 e) 8.3.25.7
			Second Serial Number	nm	54	(See reference)	7.25.1 f) 8.3.25.8
User Data	nm	193				(See reference)	7.25.1 g) 8.3.25.9

m Mandatory
nm Not mandatory (see 8.2.6)

8.3.25.7 The Second Resync Type PV field shall be coded as specified in 8.3.24.5. This PV field may only be present if the symmetric synchronize functional unit is selected.

8.3.25.8 The Second Serial Number PV field shall be coded as specified in 8.3.1.10. The length of this PV field shall not exceed the length specified by the Upper Limit Serial Number (see 8.3.4.13) and shall be limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets. This PV field may only be present if the symmetric synchronize functional unit is selected.

8.3.25.9 The User Data PV field, if present, shall contain user data supplied by the called SS-user. The length of the User Data parameter shall not exceed 512 octets if Protocol Version 1 has been selected, and shall be limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets if Protocol Version 2 is selected. If the Enclosure Item parameter is present, the User Data parameter is mandatory.

8.3.26 PREPARE (PR) SPDU

8.3.26.1 The SI field shall contain the value 7.

8.3.26.2 The parameter field shall be as specified in Table 34.

Table 34 – Parameters of the PREPARE SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Prepare Type	m	24	1 octet	7.26.1 a) 8.3.26.3
			Resync Type	nm	27	1 octet	7.26.1 b) 8.3.26.4
			Second Resync Type	nm	53	1 octet	7.26.1 c) 8.3.26.5

m Mandatory
nm Not mandatory (see 8.2.6)

8.3.26.3 The Prepare Type PV field indicates which SPDU should be expected on the transport normal flow. The value for this field shall be:

a) 1: prepare for MAJOR SYNC ACK SPDU;

b) 2: prepare for RESYNCHRONIZE SPDU;

c) 3: prepare for RESYNCHRONIZE ACK SPDU;

d) 4: prepare for ABORT SPDU.

All other values are reserved and shall not be used.

8.3.26.4 The First Resync Type PV field shall be present if and only if:

a) the symmetric synchronize functional unit is selected; and

b) the Prepare Type PV field indicates that a RESYNCHRONIZE SPDU should be expected on the transport normal flow (that is, the Prepare Type PV field has the value 2).

If this PV field is present, it shall be coded as specified in 8.3.24.5 and it shall have the same value as the First Resync Type parameter in the RESYNCHRONIZE SPDU to which this PREPARE (RESYNCHRONIZE) SPDU refers.

8.3.26.5 The Second Resync Type PV field shall be present if and only if:

a) the symmetric synchronize functional unit is selected; and

b) the Prepare Type PV field indicates that a RESYNCHRONIZE SPDU should be expected on the transport normal flow (that is, the Prepare Type PV field has the value 2).

If this PV field is present, it shall be coded as specified in 8.3.24.5 and it shall have the same value as the Second Resync Type parameter in the RESYNCHRONIZE SPDU to which this PREPARE (RESYNCHRONIZE) SPDU refers.

8.3.27 EXCEPTION REPORT (ER) SPDU

8.3.27.1 The SI field shall contain the value 0.

8.3.27.2 The parameter field shall be as specified in Table 35.

Table 35 – Parameters of the EXCEPTION REPORT SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Reflect Parameter Values	m	49	65531 octets maximum	7.27.1 8.3.27.3
m Mandatory nm Not mandatory (see 8.2.6)							

8.3.27.3 The Reflect Parameter Values PV field shall contain the bit pattern of the SPDU in error, up to and including the detected error, to a maximum of n octets

where

$$1024 \leq n \leq 65\,531$$

NOTE – Not all implementations may be able to deal with field lengths greater than 1024. It is recommended that, whenever possible, the Reflect Parameters PV field should contain the bit pattern of the SPDU in error up to and including the detected error.

8.3.28 EXCEPTION DATA (ED) SPDU

8.3.28.1 The SI field shall contain the value 48.

8.3.28.2 The parameter fields shall be as specified in Table 36.

Table 36 – Parameters of the EXCEPTION DATA SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Enclosure Item	nm	25	1 octet	7.28.1 a) 8.3.28.3
			Reason Code	m	50	1 octet	7.28.1 b) 8.3.28.4
User Data	nm	193				(See reference)	7.28.1 c) 8.3.28.5
m Mandatory nm Not mandatory (see 8.2.6)							

8.3.28.3 The Enclosure Item parameter, if present, shall be encoded as specified in 8.3.4.18. This parameter shall not be present if Protocol Version 1 is selected.

8.3.28.4 The Reason Code PV field shall contain one of the following values:

a) 0: no specific reason stated;

b) 1: temporarily unable to continue;

c) 2: reserved;

d) 3: user sequence error;

e) 4: reserved;

f) 5: local SS-user error;

g) 6: unrecoverable procedural error;

h) 128: demand data token.

All other values are reserved and shall not be used.

8.3.28.5 The User Data PV field, if present, shall contain user data supplied by the called SS-user. The length of the User Data parameter shall not exceed 512 octets if Protocol Version 1 has been selected, and shall be limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets if Protocol Version 2 is selected. If the Enclosure Item parameter is present, the User Data parameter is mandatory.

8.3.29 ACTIVITY START (AS) SPDU

8.3.29.1 The SI field shall contain the value 45.

8.3.29.2 The parameter fields shall be as specified in Table 37.

Table 37 – Parameters of the ACTIVITY START SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Enclosure Item	nm	25	1 octet	7.29.1 a) 8.3.29.3
			Activity Identifier	m	41	6 octets maximum	7.29.1 b) 8.3.29.4
User Data	nm	193				(See reference)	7.29.1 c) 8.3.29.5
m Mandatory nm Not mandatory (see 8.2.6)							

8.3.29.3 The Enclosure Item parameter, if present, shall be encoded as specified in 8.3.4.18. This parameter shall not be present if Protocol Version 1 is selected.

8.3.29.4 The Activity Identifier PV field shall be as defined by the sending SS-user.

8.3.29.5 The User Data PV field, if present, shall contain user data supplied by the called SS-user. The length of the User Data parameter shall not exceed 512 octets if Protocol Version 1 has been selected, and shall be limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets if Protocol Version 2 is selected. If the Enclosure Item parameter is present, the User Data parameter is mandatory.

8.3.30 ACTIVITY RESUME (AR) SPDU

8.3.30.1 The SI field shall contain the value 29.

8.3.30.2 The parameter fields shall be as specified in Table 38.

Table 38 – Parameters of the ACTIVITY RESUME SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Enclosure Item	nm	25	1 octet	7.30.1 a) 8.3.30.3
Linking Information	m	33	Called SS-user Reference	nm	9	64 octets maximum	7.30.1 b) 1) 8.3.30.4
			Calling SS-user Reference	nm	10	64 octets maximum	7.30.1 b) 2) 8.3.30.5
			Common Reference	nm	11	64 octets maximum	7.30.1 b) 3) 8.3.30.6
			Additional Reference Information	nm	12	4 octets maximum	7.30.1 b) 4) 8.3.30.7
			Old Activity Identifier	m	41	6 octets maximum	7.30.1 b) 5) 8.3.30.8
			Serial Number	m	42	(See reference)	7.30.1 b) 6) 8.3.30.9
			Second Serial Number	nm	54	(See reference)	7.30.1 b) 7) 8.3.30.10
			New Activity Identifier	m	41	6 octets maximum	7.30.1 c) 8.3.30.11
User Data	nm	193				(See reference)	7.30.1 d) 8.3.30.12

m Mandatory
nm Not mandatory (see 8.2.6)

8.3.30.3 The Enclosure Item parameter, if present, shall be encoded as specified in 8.3.4.18. This parameter shall not be present if Protocol Version 1 is selected.

8.3.30.4 The Called SS-user Reference PV field shall be as defined by the sending SS-user.

8.3.30.5 The Calling SS-user Reference PV field shall be as defined by the sending SS-user.

8.3.30.6 The Common Reference PV field shall be as defined by the sending SS-user.

8.3.30.7 The Additional Reference Information PV field shall be as defined by the sending SS-user.

8.3.30.8 The Old Activity Identifier PV field shall be as defined by the sending SS-user.

8.3.30.9 The Serial Number PV field shall be coded as specified in 8.3.1.10. The length of this PV field shall not exceed the length specified by the Upper Limit Serial Number (see 8.3.4.13) and shall be limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets.

8.3.30.10 The Second Serial Number PV field shall be coded as specified in 8.3.1.10. The length of this PV field shall not exceed the length specified by the Upper Limit Serial Number (see 8.3.4.13) and shall be limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets. This PV field may only be present if the symmetric synchronize functional unit is selected.

8.3.30.11 The New Activity Identifier PV field shall be as defined by the sending SS-user.

8.3.30.12 The User Data PV field, if present, shall contain user data supplied by the called SS-user. The length of the User Data parameter shall not exceed 512 octets if Protocol Version 1 has been selected, and shall be limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets if Protocol Version 2 is selected. If the Enclosure Item parameter is present, the User Data parameter is mandatory.

8.3.31 ACTIVITY INTERRUPT (AI) SPDU

8.3.31.1 The SI field shall contain the value 25.

8.3.31.2 The parameter fields shall be as specified in Table 39.

Table 39 – Parameters of the ACTIVITY INTERRUPT SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Enclosure Item	nm	25	1 octet	7.31 a) 8.3.31.3
			Reason Code	nm	50	1 octet	7.31.1 b) 8.3.31.4
User Data	nm	193				(See reference)	7.31.1 c) 8.3.31.5
m Mandatory nm Not mandatory (see 8.2.6)							

8.3.31.3 The Enclosure Item parameter, if present, shall be encoded as specified in 8.3.4.18. This parameter shall not be present if Protocol Version 1 is selected.

8.3.31.4 The Reason Code PV field shall contain one of the following values:

a) 0: no specific reason stated;

b) 1: temporarily unable to continue;

c) 2: reserved;

d) 3: user sequence error;

e) 4: reserved;

f) 5: local SS-user error;

g) 6: unrecovable procedural error;

h) 128: demand data token.

All other values are reserved and shall not be used.

8.3.31.5 The User Data PV field, if present, shall contain user data supplied by the SS-user. This PGI unit shall not be present if Protocol Version 1 is selected. The length of the User Data parameter is limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets.

8.3.32 ACTIVITY INTERRUPT ACK (AIA) SPDU

8.3.32.1 The SI field shall contain the value 26.

8.3.32.2 The parameter fields shall be as specified in Table 40.

Table 40 – Parameters of the ACTIVITY INTERRUPT ACK SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Enclosure Item	nm	25	1 octet	7.32.1 a) 8.3.32.3
User Data	nm	193				(See reference)	7.32.1 b) 8.3.32.4
m Mandatory nm Not mandatory (see 8.2.6)							

8.3.32.3 The Enclosure Item parameter, if present, shall be encoded as specified in 8.3.4.18. This parameter shall not be present if Protocol Version 1 is selected.

8.3.32.4 The User Data PV field, if present, shall contain user data supplied by the SS-user. This PGI unit shall not be present if Protocol Version 1 is selected. The length of the User Data parameter is limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets.

8.3.33 ACTIVITY DISCARD (AD) SPDU

8.3.33.1 The SI field shall contain the value 57.

8.3.33.2 The parameter fields shall be as specified in Table 41.

Table 41 – Parameters of the ACTIVITY DISCARD SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Enclosure Item	nm	25	1 octet	7.33.1 a) 8.3.33.3
			Reason Code	nm	50	1 octet	7.33.1 b) 8.3.33.4
User Data	nm	193				(See reference)	7.33.1 c) 8.3.33.5
m Mandatory nm Not mandatory (see 8.2.6)							

8.3.33.3 The Enclosure Item parameter, if present, shall be encoded as specified in 8.3.4.18. This parameter shall not be present if Protocol Version 1 is selected.

8.3.33.4 The Reason Code PV field shall contain one of the following values:

a) 0: no specific reason stated;

b) 1: temporarily unable to continue;

c) 2: reserved;

d) 3: user sequence error;

e) 4: reserved;

f) 5: local SS-user error;

g) 6: unrecovable procedural error;

h) 128: demand data token.

All other values are reserved and shall not be used.

8.3.33.5 The User Data PV field, if present, shall contain user data supplied by the SS-user. This PGI unit shall not be present if Protocol Version 1 is selected. The length of the User Data parameter is limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets.

8.3.34 ACTIVITY DISCARD ACK (ADA) SPDU

8.3.34.1 The SI field shall contain the value 58.

8.3.34.2 The parameter fields shall be as specified in Table 42.

Table 42 – Parameters of the ACTIVITY DISCARD ACK SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Enclosure Item	nm	25	1 octet	7.34.1 a) 8.3.34.3
User Data	nm	193				(See reference)	7.34.1 b) 8.3.34.4
m Mandatory nm Not mandatory (see 8.2.6)							

8.3.34.3 The Enclosure Item parameter, if present, shall be encoded as specified in 8.3.4.18. This parameter shall not be present if Protocol Version 1 is selected.

8.3.34.4 The User Data PV field, if present, shall contain user data supplied by the SS-user. This PGI unit shall not be present if Protocol Version 1 is selected. The length of the User Data parameter is limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets.

8.3.35 ACTIVITY END (AE) SPDU

8.3.35.1 The SI field shall contain the value 41. This is the same value as the SI field for the MAJOR SYNC POINT SPDU (see 8.3.22).

8.3.35.2 The parameter fields shall be as specified in Table 43.

Table 43 – Parameters of the ACTIVITY END SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Enclosure Item	nm	25	1 octet	7.36.1 a) 8.3.35.3
			Serial Number	m	42	(See reference)	7.36.1 b) 8.3.35.4
User Data	nm	193				(See reference)	7.36.1 c) 8.3.35.5
m Mandatory nm Not mandatory (see 8.2.6)							

8.3.35.3 The Enclosure Item parameter, if present, shall be encoded as specified in 8.3.4.18. This parameter shall not be present if Protocol Version 1 is selected.

8.3.35.4 The Serial Number PV field shall be coded as specified in 8.3.1.10. The length of this PV field shall not exceed the length specified by the Upper Limit Serial Number (see 8.3.4.13) and shall be limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets.

8.3.35.5 The User Data PV field, if present, shall contain user data supplied by the called SS-user. The length of the User Data parameter shall not exceed 512 octets if Protocol Version 1 has been selected, and shall be limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets if Protocol Version 2 is selected. If the Enclosure Item parameter is present, the User Data parameter is mandatory.

8.3.36 ACTIVITY END ACK (AEA) SPDU

The ACTIVITY END ACK SPDU is identical to the MAJOR SYNC ACK SPDU (see 8.3.23).

8.4 Additional encoding rules for segmented SSDUs

- ACCEPT SPDU;
- REFUSE SPDU;
- FINISH SPDU;
- DISCONNECT SPDU;
- NOT FINISHED SPDU;
- ABORT SPDU;
- CAPABILITY DATA SPDU;
- CAPABILITY DATA ACK SPDU;
- GIVE TOKENS SPDU;
- PLEASE TOKENS SPDU;
- GIVE TOKENS CONFIRM SPDU;
- MINOR SYNC POINT SPDU;
- MINOR SYNC ACK SPDU;
- MAJOR SYNC POINT SPDU;
- MAJOR SYNC ACK SPDU;
- RESYNCHRONIZE SPDU;
- RESYNCHRONIZE ACK SPDU;
- EXCEPTION DATA SPDU;
- ACTIVITY START SPDU;
- ACTIVITY RESUME SPDU;
- ACTIVITY INTERRUPT SPDU;
- ACTIVITY INTERRUPT ACK SPDU;
- ACTIVITY DISCARD SPDU;
- ACTIVITY DISCARD ACK SPDU;
- ACTIVITY END SPDU;
- ACTIVITY END ACK SPDU.

8.4.1 First SPDU in sequence

The first SPDU in the sequence shall be as specified in 8.3.

8.4.2 Subsequent SPDUs in a sequence

8.4.2.1 For all SPDUs except the REFUSE SPDU and the MINOR SYNC ACK SPDU, the encoding shall be as follows.

8.4.2.1.1 The SI field shall have the same value as the SI field value of the initial SPDU of the sequence.

8.4.2.1.2 The parameter fields shall be as specified in Table 44.

Table 44 – Parameters of subsequent SPDUs when segmenting is required

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Enclosure Item	m	25	1 octet	7.37.1 8.4.2.1.3
User Data	nm	193				(See reference)	7.37.1 8.4.2.1.4
m Mandatory nm Not mandatory (see 8.2.6)							

8.4.2.1.3 The Enclosure Item PV field shall indicate whether or not this SPDU is the end of the SSDU. The encoding shall be:

a) bit 1 = 0: not beginning of SSDU;

b) bit 2 = 1: end of SSDU;
bit 2 = 0: not end of SSDU.

Bits 3-8 are reserved.

8.4.2.1.4 The User Data field, if present, shall contain a segment of the associated SSDU. The User Data field shall be present if the Enclosure Item has bit 2 = 0.

8.4.2.2 For the REFUSE SPDU the encoding shall be as follows.

8.4.2.2.1 The SI field shall have the same value as the SI field value of the initial SPDU of the sequence.

8.4.2.2.2 The parameter fields shall be as specified in Table 45.

Table 45 – Parameters of subsequent REFUSE SPDUs when segmenting is required

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Enclosure Item	m	25	1 octet	7.37.1 8.4.2.2.3
			Reason Code	nm	50	65528 octets maximum	7.37.1 8.4.2.2.4
m Mandatory nm Not mandatory (see 8.2.6)							

8.4.2.2.3 The Enclosure Item PV field shall indicate whether or not this SPDU is the end of the SSDU. The encoding shall be:

a) bit 1 = 0: not beginning of SSDU;

b) bit 2 = 1: end of SSDU;
bit 2 = 0: not end of SSDU.

Bits 3-8 are reserved.

8.4.2.2.4 The Reason Code field, if present, shall contain a segment of the associated SSDU. The Reason Code field shall be present if the Enclosure Item has bit 2 = 0.

8.4.2.3 For the MINOR SYNC ACK SPDU the encoding shall be as follows.

8.4.2.3.1 The SI field shall have the same value as the SI field value of the initial SPDU of the sequence.

8.4.2.3.2 The parameter fields shall be as specified in Table 46.

Table 46 – Parameters of subsequent MINOR SYNC ACK SPDUs when segmenting is required

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Enclosure Item	m	25	1 octet	7.37.1 8.4.2.3.3
			User Data	nm	46	65528 octets maximum	7.37.1 8.4.2.3.4
m Mandatory nm Not mandatory (see 8.2.6)							

8.4.2.3.3 The Enclosure Item PV field shall indicate whether or not this SPDU is the end of the SSDU. The encoding shall be:

a) bit 1 = 0: not beginning of SSDU;

b) bit 2 = 1: end of SSDU;
bit 2 = 0: not end of SSDU.

Bits 3-8 are reserved.

8.4.2.3.4 The User Data field, if present, shall contain a segment of the associated SSDU. The User Data field shall be present if the Enclosure Item has bit 2 = 0.

SECTION 3 – CONFORMANCE

9 Conformance to this Recommendation | International Standard

9.1 Static conformance requirements

9.1.1 A system claiming conformance to this Recommendation | International Standard shall exhibit external behaviour consistent with having implemented an SPM for the kernel functional unit together with either or both of the half-duplex and the duplex functional units.

9.1.2 The system may exhibit external behaviour consistent with containing an implementation of any other functional unit provided that:

a) if the capability data functional unit is implemented, the activity management functional unit shall also be implemented; and

b) if the exceptions functional unit is implemented, the half-duplex functional unit shall also be implemented.

9.1.3 Claims of conformance shall state:

a) which functional units are implemented;

b) whether or not extended concatenation is implemented;

c) whether or not segmenting is implemented and, if segmenting is implemented, the maximum size of TSDU which the system is capable of handling;

d) whether or not the use of the transport expedited service is implemented;

e) which protocol versions are implemented;

f) any limit imposed by an implementation on the number of octets of SS-user data which can be passed in a single session service primitive.

9.2 Dynamic conformance requirements

9.2.1 For all protocol versions which are claimed to be implemented, the system shall be capable of initiating a session connection (by sending a CONNECT SPDU) or responding to a CONNECT SPDU (according to the procedures specified in clause 7), or both.

9.2.2 The system shall also follow:

a) all the remaining procedures in the kernel functional unit; and

b) all the procedures for each functional unit that the system claims to implement,

where following the procedures specified in a) and b) shall mean:

c) accepting all valid sequences of SPDUs received from peer equipment; any action taken in response shall not violate the procedures specified in this Recommendation | International Standard;

d) responding to all incorrect sequences of SPDUs received for a defined state of a session connection without violating the procedures specified in this Recommendation | International Standard.

9.3 Protocol Implementation Conformance Statement

9.3.1 The implementor shall provide a Protocol Implementation Conformance Statement (PICS) by completing the PICS proforma provided in ITU-T Rec. X.245 | ISO/IEC 8327-2 or a compliant PICS proforma provided by a test laboratory.

9.3.2 A PICS proforma which conforms to this Recommendation | International Standard shall preserve the technical content and maintain the same sequence of items as ITU-T Rec. X.245 | ISO/IEC 8327-2.

Annex A

State tables

(This annex forms an integral part of this Recommendation | International Standard)

A.1 General

This annex describes the session protocol in terms of state tables. The state tables show the state of a session connection, the events that occur in the protocol, the actions taken and the resultant state.

These state tables do not constitute a formal definition of the session protocol; they are included to provide a more precise specification of the elements of procedure described in clause 7. In case of arbitration or dispute, this annex takes precedence over clause 7.

Table A.1 specifies the abbreviated name, category and name of each incoming event. The categories are SS-user event, TS-provider event, timer event and valid SPDU event.

Table A.2 specifies the abbreviated name and name of each state.

Table A.3 specifies the abbreviated name, category and name of each outgoing event. The categories are SS-provider event, TS-user event and SPDU event.

Table A.4 summarizes the operations on the variables V(A), V(M), V(R) and Vsc when the symmetric synchronize functional unit has not been selected.

Table A.5 summarizes the operations on the variables V(A), V(M), V(R) and Vsc when the symmetric synchronize functional unit has been selected.

Table A.6 specifies the specific actions.

Table A.7 specifies the predicates.

Tables A.8 to A.16 specify the state tables when the symmetric synchronize functional unit has not been selected.

Tables A.17 to A.25 specify the state tables when the symmetric synchronize functional unit has been selected.

A.2 Notation for state tables

A.2.1 Incoming events, states and outgoing events are represented by their abbreviated names.

A.2.2 Specific actions are represented by the notation [n], where n is the number of the specific action in Table A.6.

A.2.3 Notes are represented by the notation (n), where n is the number of the Note at the foot of Table A.7.

A.2.4 Predicates are represented by the notation pn, where n is the number of the predicate in Table A.7.

A.2.5 Boolean operators are represented by the following notation:

&	AND
¬	NOT
OR	OR

A.3 Conventions for entries in state tables

A.3.1 The intersection of each state and incoming event which is invalid is left blank.

A.3.2 The intersection of each state and incoming event which is valid contains entries which are either:

a) an action list which:

1) may contain outgoing events and/or specific actions;

2) always contains the resultant state; or

b) one or more conditional action lists, each consisting of:

1) a predicate expression comprising predicates and boolean operators;

2) an action list [as in A.3.2 a)].

NOTE – the action lists and conditional action lists use the notation in A.2.

A.3.3 The intersection of each state and incoming event which is not logically possible for the SPM is indicated by // in the top lefthand corner of the intersection.

NOTE – Such entries are a consequence of the tabular presentation of the state tables.

A.4 Actions to be taken by the SPM

The state tables define the action to be taken by the SPM.

A.4.1 Invalid intersections

If the intersection of the state and an incoming event is invalid, one of the following actions shall be taken.

A.4.1.1 If the incoming event comes from the SS-user, any action taken by the SPM is a local matter.

A.4.1.2 If the incoming event is related to a received SPDU and if the state of the transport connection makes it possible, the SPM shall either:

a) take the following actions:

1) issue an S-P-ABORT indication;

2) send an ABORT SPDU;

3) start the timer, TIM;

4) wait for a T-DISCONNECT indication or an ABORT ACCEPT SPDU (STA 16); or

b) if the following conditions hold:

1) the data token is available but not assigned to the SPM; and

2) – the activity management functional unit has not been selected; or

– the activity management functional unit has been selected and an activity is in progress; or

– the activity management functional unit has been selected and the SPM is in STA 22; and

3) the exceptions functional unit has been selected; and

4) the session connection is in the data transfer phase (i.e. states 4A, 4B, 5A, 5B, 5C, 6, 10A, 10B, 11A, 11B, 11C, 15A, 15B, 15C, 19, 20, 22, 713);

take the following actions:

5) send an EXCEPTION REPORT SPDU;

6) issue an S-P-EXCEPTION-REPORT indication;

7) enter STA 20 and wait for a recovery request or SPDU.

NOTE – It should be noted that sending an EXCEPTION REPORT SPDU may lead to an SPM deadlock. It is therefore advised to send the ABORT SPDU rather than the EXCEPTION REPORT SPDU, especially in the case of protocol errors.

A.4.1.3 If the incoming event falls into neither of the above categories (including those which are impossible by the definition of the behaviour of the SPM or TS-provider) any action taken by the SPM is a local matter.

A.4.2 Valid intersections

If the intersection of the state and incoming event is valid, one of the following actions shall be taken.

A.4.2.1 If the intersection contains an action list, the SPM shall take the specific actions in the order specified in the state tables.

A.4.2.2 If the intersection contains one or more conditional action lists, for each predicate expression that is true the SPM shall take the specific actions in the order given in the action list associated with the predicate expression. If none of the predicate expressions are true, the SPM shall take one of the actions defined in A.4.1.

A.4.2.3 The state tables do not take account of segmented SSDUs. When an outgoing SSDU is to be segmented or an incoming SSDU is segmented, the procedures defined in 7.37 apply to the outgoing event at the appropriate intersection of the state tables (that part of the action which transmits the SPDU).

A.4.3 Receipt of SPDUs

A.4.3.1 Valid SPDUs

The SPM shall process valid SPDUs as specified in Tables A.8 to A.25.

A.4.3.1.1 Rules for extensibility

This Recommendation | International Standard does not specify the action to be taken in response to a PGI unit containing a PGI code listed in Annex B, or to a PI unit containing a PI code listed in Annex B.

An SPM receiving an SPDU containing a valid SI field but containing a PGI unit whose PGI code is not specified in 8.3 or in Annex B, shall ignore that PGI unit (see Notes).

An SPM receiving an SPDU containing a valid SI field but containing a PI unit whose PI code is not specified in 8.3 or in Annex B, shall ignore that PI unit (see Notes).

The SPM shall ignore any bits within a parameter field which are specified as reserved in 8.3.

NOTES

1 The received SPDU is processed as though the unknown PGI and/or PI units were not present in the SPDU.

2 These provisions permit communication with systems operating other versions of this protocol.

A.4.3.1.2 User Data length restrictions

If an SPM receives an SPDU, or an ordered sequence of SPDUs which together comprise a single SSDU, which contains more SS-user data than the SPM is prepared to accept (and as stated in the PICS), it shall take the actions defined in either A.4.1.2 a) or A.4.1.2 b).

A.4.3.2 Invalid SPDUs

If an invalid SPDU is received, the SPM shall:

a) take the actions defined in A.4.1.2 a); or

b) take the actions defined in A.4.1.2 b); or

c) take any other action that does not violate the procedures specified in this Recommendation | International Standard; or

d) take no action.

A.5 Definitions of sets and variables

The following sets and variables are specified in this Recommendation | International Standard.

A.5.1 Functional units

The set of all functional units specified in this Recommendation | International Standard is defined as:

fu-dom = {FD, HD, EXCEP, TD, NR, SY, SS, DS, MA, RESYN, EX, ACT, CD}

where

FD	Duplex functional unit
HD	Half-duplex functional unit
EXCEP	Exceptions functional unit
TD	Typed data functional unit
NR	Negotiated release functional unit
SY	Minor synchronize functional unit
SS	Symmetric synchronize functional unit
DS	Data separation functional unit
MA	Major synchronize functional unit
RESYN	Resynchronize functional unit
EX	Expedited data functional unit
ACT	Activity management functional unit
CD	Capability data exchange functional unit

A boolean function FU is defined over fu-dom as follows:

for f in fu-dom

FU(f) = true if and only if the functional unit f has been selected during the session connection establishment phase.

The value is set when the ACCEPT SPDU is sent or received.

A.5.2 Tokens

The set of all tokens specified in this Recommendation | International Standard is defined as:

tk-dom = {mi, ma, tr, dk}

where

mi is the synchronize-minor token;

ma is the major/activity token;

tr is the release token;

dk is the data token.

The following boolean functions are defined over tk-dom:

a) AV(t), for t in tk-dom, is a function which defines the availability of the corresponding token and has the following values:

AV(mi) = FU(SY);

AV(dk) = FU(HD);

AV(tr) = FU(NR);

AV(ma) = FU(MA) OR FU(ACT).

b) OWNED(t), for t in tk-dom, is a function which defines the assignment of the corresponding token and is defined as:

OWNED(t) = true: if token assigned to the SPM;

OWNED(t) = false: if token not assigned to the SPM.

OWNED(t) is not defined if AV(t) = false. OWNED(t) is set when:

1) the ACCEPT SPDU is sent or received; or
2) the RESYNCHRONIZE ACK SPDU is sent or received; or
3) the GIVE TOKENS SPDU is sent or received; or
4) the GIVE TOKENS CONFIRM SPDU is sent or received;
5) the ACTIVITY INTERRUPT ACK SPDU is sent or received;
6) the ACTIVITY DISCARD ACK SPDU is sent or received.

c) I(t), for t in tk-dom, is a function which, when true, indicates that the SPM has Initiating rights for the behaviour controlled by the token. This applies even if the corresponding token is not available:

$$I(t) = \neg AV(t) \text{ OR } OWNED(t)$$

d) A(t), for t in tk-dom, is a function which, when true, indicates that the SPM has Accepting rights for the behaviour controlled by the token. This applies even if the corresponding token is not available:

$$A(t) = \neg AV(t) \text{ OR } \neg OWNED(t)$$

e) II(t), for t in tk-dom, is a function which, when true, indicates that the SPM has Initiating rights as I(t), but this applies to the case when the behaviour may only be initiated if the corresponding token is available and owned:

$$II(t) = AV(t) \text{ AND } OWNED(t)$$

f) AA(t), for t in tk-dom, is a function which, when true, indicates that the SPM has Accepting rights as A(t), but only if the corresponding token is available, but not owned:

$$AA(t) = AV(t) \text{ AND } \neg OWNED(t)$$

A.5.3 SET of tokens

The following subsets of tk-dom are defined:

RT = {tokens requested in the input event};

GT = {tokens given in the input event}.

For the purpose of the following function definitions, two further sets are defined:

F = {AV, OWNED, I, A, II, AA}(the set of functions defined in A.5.2);

S = the set of subsets of tk-dom.

The following functions are defined over F and S:

a) ALL(f, s), for f in F and s in S:

ALL(f, s) = true: all of the f(t) for t in s are true or s is empty;

For example:

ALL(A, tk-dom) = true: none of the available tokens are owned (e.g. on receipt of a FINISH SPDU)

b) ANY(f, s), for f in F and s in S:

ANY(f, s) = true: any f(t) = true for t in s and s is not empty;

For example:

ANY(II, tk-dom) = true: at least one of the available tokens is owned.

A.5.4 Variables

A.5.4.1 TEXP

TEXP is a boolean variable having the following values:

TEXP = true: use of transport expedited service is selected for use on this session connection;

TEXP = false: use of transport expedited service is not selected for use on this session connection.

A.5.4.2 Vact

Vact is a boolean variable having the following values when the activity management functional unit has been selected (FU(ACT) = true):

Vact = true: an activity is in progress;

Vact = false: no activity is in progress.

Vact has no defined value if FU(ACT) = false.

Vact is set as follows:

a) Vact is set false during the connection establishment phase, if the activity management functional unit has been selected (FU(ACT) = true); otherwise, Vact is not set;

b) Vact is set true when the ACTIVITY START SPDU or ACTIVITY RESUME SPDU is sent or received (only possible when FU(ACT) = true);

c) Vact is set false when the ACTIVITY DISCARD ACK SPDU or ACTIVITY INTERRUPT ACK SPDU is sent or received;

d) Vact is set to Vnextact when a MAJOR SYNC ACK SPDU or an ACTIVITY END ACK SPDU is sent or received.

A.5.4.3 Vnextact

Vnextact is a boolean variable which is used when the activity management functional unit has been selected (FU(ACT) = true). It is used to indicate the next value of Vact when a MAJOR SYNC ACK SPDU or an ACTIVITY END ACK SPDU is sent or received. Vnextact is set when a MAJOR SYNC POINT SPDU or an ACTIVITY END SPDU is sent or received:

a) Vnextact is set false if FU(ACT) = true and an ACTIVITY END SPDU is sent or received;

b) Vnextact is set true if FU(ACT) = true and a MAJOR SYNC POINT SPDU is sent or received.

Vnextact has no defined value if FU (ACT) = false.

A.5.4.4 Vrsp and Vrspnb

These variables are used to resolve resynchronization collisions when the symmetric synchronize functional unit has not been selected.

Vrsp indicates what kind of resynchronization is currently in progress:

Vrsp = no: no resynchronization in progress;

Vrsp = a: resynchronize abandon;

Vrsp = r: resynchronize restart;

Vrsp = s: resynchronize set;

Vrsp = dsc: discard activity;

Vrsp = int: interrupt activity.

Vrspnb indicates the serial number in the case of resynchronize restart.

Vrsp and, if necessary Vrspnb, are set when a RESYNCHRONIZE SPDU, ACTIVITY INTERRUPT SPDU or an ACTIVITY DISCARD SPDU is sent or received. Vrsp is set to no when the SPM goes to STA 713.

A.5.4.5 Vrsps, Vrspr, Vspnbs, and Vrspnbr

When symmetric synchronization is used, Vrspnbr indicates the serial number for the SPM's receiving flow and Vrspnbs indicates the serial number for the SPM's sending flow in the case of resynchronize restart. Vrspr indicates the type of resynchronize (a, s, r, or no) for the receiving flow; Vrsps indicates the type of resynchronize for the sending flow.

Vrsps and Vrspr and, if necessary Vrspnbs and Vrspnbr, are set when a RESYNCHRONIZE SPDU, ACTIVITY INTERRUPT SPDU, or an ACTIVITY DISCARD SPDU is sent or received. In the case of a collision, the variables are updated to contain the prevailing values (see A.5.4.6.2).

A.5.4.6 SPMwinner

When a resynchronization collision has been detected, the boolean function SPMwinner is calculated to determine which SPM wins against the colliding event. Different synchronization variables are used to evaluate SPMwinner depending on whether or not the symmetric synchronize functional unit has been selected.

A.5.4.6.1 SPMwinner without symmetric synchronization

SPMwinner is a boolean function which is used during resynchronization collision, that is when:

a) a RESYNCHRONIZE SPDU is received and Vrsp is not equal to no;

b) an S-RESYNCHRONIZE request is received and Vrsp is not equal to no.

The SPMwinner condition is true if the SPM (which holds the current resynchronization) wins against the new colliding event.

The SPMwinner condition is calculated as follows:

a) The next Vrsp and Vrspnb values are evaluated according to the parameters of the received event. The new calculated value for Vrsp is compared to the current Vrsp with the following ordering rule:

dsc prevails over int;

int prevails over a;

a prevails over s;

s prevails over r.

If both are equal to r, then the new calculated value for Vrspnb is compared to the current value of Vrspnb and the lower value prevails.

b) If the current value of Vrsp (and Vrspnb if necessary) prevails, then the SPMwinner condition is true (in this case, the current resynchronization prevails over the colliding one).

c) If the current value of Vrsp (and Vrspnb if necessary) does not prevail, then the SPMwinner condition is false (in this case, the colliding resynchronization prevails over the current one).

d) If the above comparison results in equality and if the colliding event has been generated by the initiator of the session connection (either a RESYNCHRONIZE SPDU was received from the session connection initiator or a local S-RESYNCHRONIZE request was issued by the session connection initiator), then the SPMwinner condition is false.

If the SPM is winner (SPMwinner condition is true), then the current resynchronization wins against the colliding one and Vrsp and Vrspnb are not updated.

If the SPM is not winner (SPMwinner condition is false), then the colliding resynchronization is taken into account and Vrsp and Vrspnb are updated.

A.5.4.6.2 SPMwinner with symmetric synchronization

When the symmetric synchronize functional unit has been selected, the SPMwinner condition is calculated as follows:

a) The resynchronize type and serial number values are evaluated for each direction of flow according to the parameters of the received event. The new calculated value for Vrsps (or Vrspr) is compared to the current Vrsps (or Vrspr) with the following ordering rule:

dsc prevails over int;

int prevails over a;

a prevails over s;

s prevails over r;

r prevails over no.

If both are equal to r for a direction of flow, then the new calculated value for Vrspnbs (or Vrspnbr) is compared to the current value and the lower value prevails.

b) If the current values of Vrsps, Vrspr, Vrspnbs and/or Vrspnbr all prevail, then the SPMwinner condition is true (in this case, the current resynchronization prevails over the colliding one).

c) If the colliding event values of Vrsps, Vrspr, Vrspnbs and/or Vrspnbr all prevail, then the SPMwinner condition is false (in this case, the colliding resynchronization prevails over the current one).

d) Otherwise:

1) if the colliding event has been generated by the initiator of the session connection, then the SPMwinner condition is false;

2) if the colliding event has been generated by the acceptor of the session connection, then the SPMwinner condition is true.

If the SPM is winner (SPMwinner condition is true), then the current resynchronization wins against the colliding one.

If the SPM is not winner (SPMwinner condition is false), then the colliding resynchronization wins.

In all cases, Vrsps, Vrspr, Vrspnbs and Vrspnbr are updated to reflect the prevailing values.

A.5.4.7 Vtca

Vtca is a boolean variable having the following values:

Vtca = false: the SPM initiated the T-CONNECT request (transport connection initiator);

Vtca = true: the SPM received the T-CONNECT indication (transport connection acceptor).

A.5.4.8 Vtrr

Vtrr is a boolean variable having the following values:

Vtrr = true: the transport connection can be reused by the SPM for another session connection;

Vtrr = false: the transport connection cannot be reused by the SPM for another session connection.

A.5.4.9 Vcoll

Vcoll is a boolean variable having the following values:

Vcoll = true: a collision of FINISH SPDUs has been detected;

Vcoll = false: there has not been a collision of FINISH SPDUs.

This variable is set false during the session connection establishment phase.

A.5.4.10 Vdnr

Vdnr is a boolean variable having the following values:

Vdnr = true: a DISCONNECT SPDU has been received in STA09 (following a collision of FINISH SPDUs);

Vdnr = false: no DISCONNECT SPDU has been received.

This variable is set to false during the connection establishment phase.

A.5.4.11 V(A)

When the symmetric synchronize functional unit has not been selected, V(A) is used by the SPM and is the lowest serial number to which a synchronization point confirmation is expected. No confirmation is expected when V(A) = V(M).

A.5.4.12 V(M)

When the symmetric synchronize functional unit has not been selected, V(M) is used by the SPM and is the next serial number to be used.

A.5.4.13 V(R)

When the symmetric synchronize functional unit has not been selected, V(R) is used by the SPM and is the lowest serial number to which resynchronization restart is permitted.

A.5.4.14 Vsc

Vsc is a boolean variable having the following values:

Vsc = true: the SS-user has the right to issue minor synchronization point responses when V(A) is less than V(M);

Vsc = false: the SS-user does not have the right to issue minor synchronization point responses.

Vsc is set false during the connection establishment phase and when a MINOR SYNC POINT SPDU is sent. Vsc is set true when a MINOR SYNC POINT SPDU is received.

Vsc is not used if the symmetric synchronize functional unit has been selected.

A.5.4.15 V(Ado)

V(Ado) is used by the SPM and is the highest synchronization point serial number which was sent in a MINOR SYNCHRONIZATION POINT SPDU with the data separation parameter set to true. Received SPDUs cannot be discarded in case of resynchronization when V(Ado) is greater than or equal to V(A).

A.5.4.16 V(Adi)

V(Adi) is used by the SPM and is the highest synchronization point serial number which was received in a MINOR SYNCHRONIZATION POINT SPDU with the data separation parameter set to true.

A.5.4.17 V(As), V(Ar)

V(As) and V(Ar) are used by the SPM to manage symmetric synchronization point confirmations.

V(As) is the lowest serial number on the SPM's sending data flow to which a synchronization point confirmation is expected to be received. No confirmation is expected to be received when V(As) = V(Ms).

V(Ar) is the lowest serial number on the SPM's receiving data flow for which a confirmation has not yet been sent. No confirmation will be sent by the SPM when V(Ar) = V(Mr).

A.5.4.18 V(Ms), V(Mr)

V(Ms) and V(Mr) are used by the SPM to maintain the next symmetric synchronization serial number to be used on the sending and receiving data flows respectively.

V(Ms) is the serial number of the next synchronization point to be sent. V(Mr) is the serial number of the next synchronization point to be received.

A.5.4.19 V(Rs), V(Rr)

V(Rs) and V(Rr) are used by the SPM to maintain the lowest serial numbers to which resynchronize restart is permitted.

V(Rs) is the lowest serial number on the SPM's sending data flow to which resynchronize restart is permitted.

V(Rr) is the lowest serial number on the SPM's receiving data flow to which resynchronize restart is permitted.

A.5.4.20 Discard-rcv-flow – Discard-snd-flow

Discard-rcv-flow and Discard-snd-flow are boolean variables used by the SPM to maintain whether the receiving and sending flows, respectively, are in the process of being resynchronized. If true, the appropriate flow is in the resynchronize process.

When the resynchronize functional unit has been selected without the symmetric synchronize functional unit, resynchronization on a single direction of flow is not permitted. In this case, both boolean variables are always set when resynchronization is in progress.

NOTE – Table A.4 summarizes the operations on V(A), V(M), V(R), and Vsc when the symmetric synchronize functional unit has not been selected. Table A.5 summarizes the operations on V(As), V(Ar), V(Ms), V(Mr), V(Rs), and V(Rr) when symmetric synchronization is used.

Table A.1 – Incoming events

Abbreviated name	Category	Name and description
SACTDreq	SS-user	S-ACTIVITY-DISCARD request primitive
SACTDrsp	SS-user	S-ACTIVITY-DISCARD response primitive
SACTEreq	SS-user	S-ACTIVITY-END request primitive
SACTErsp	SS-user	S-ACTIVITY-END response primitive
SACTIreq	SS-user	S-ACTIVITY-INTERRUPT request primitive
SACTIrsp	SS-user	S-ACTIVITY-INTERRUPT response primitive
SACTRreq	SS-user	S-ACTIVITY-RESUME request primitive
SACTSreq	SS-user	S-ACTIVITY-START request primitive
SCDreq	SS-user	S-CAPABILITY-DATA request primitive
SCDrsp	SS-user	S-CAPABILITY-DATA response primitive
SCGreq	SS-user	S-CONTROL-GIVE request primitive
SCONreq	SS-user	S-CONNECT request primitive
SCONrsp+	SS-user	S-CONNECT (accept) response primitive
SCONrsp–	SS-user	S-CONNECT (reject) response primitive
SDTreq	SS-user	S-DATA request primitive
SEXreq	SS-user	S-EXPEDITED-DATA request primitive
SGTreq	SS-user	S-TOKEN-GIVE request primitive
SPTreq	SS-user	S-TOKEN-PLEASE request primitive
SRELreq	SS-user	S-RELEASE request primitive
SRELrsp+	SS-user	S-RELEASE (accept) response primitive
SRELrsp–	SS-user	S-RELEASE (reject) response primitive
SRSYNreq(a)	SS-user	S-RESYNCHRONIZE (abandon) request primitive
SRSYNreq(r)	SS-user	S-RESYNCHRONIZE (restart) request primitive
SRSYNreq(s)	SS-user	S-RESYNCHRONIZE (set) request primitive
SRSYNrsp	SS-user	S-RESYNCHRONIZE response primitive
SSYNMreq	SS-user	S-SYNC-MAJOR request primitive
SSYNMrsp	SS-user	S-SYNC-MAJOR response primitive
SSYNmreq	SS-user	S-SYNC-MINOR request primitive
SSYNmdreq	SS-user	S-SYNC-MINOR (data separation) request primitive
SSYNmrsp	SS-user	S-SYNC-MINOR response primitive
STDreq	SS-user	S-TYPED-DATA request primitive
SUABreq	SS-user	S-U-ABORT request primitive
SUERreq	SS-user	S-U-EXCEPTION-REPORT request primitive
TCONind	TS-provider	T-CONNECT indication primitive
TCONcnf	TS-provider	T-CONNECT confirm primitive
TDISind	TS-provider	T-DISCONNECT indication primitive
TIM Timer	Time out	

Table A.1 (*concluded*) – Incoming events

Abbreviated name	Category	Name and description
AA	SPDU	ABORT ACCEPT SPDU
AB-nr	SPDU	ABORT (not reuse) SPDU
AB-r	SPDU	ABORT (reuse) SPDU
AC	SPDU	ACCEPT SPDU (Note 1)
AD	SPDU	ACTIVITY DISCARD SPDU
ADA	SPDU	ACTIVITY DISCARD ACK SPDU
AE	SPDU	ACTIVITY END SPDU
AEA	SPDU	ACTIVITY END ACK SPDU
AI	SPDU	ACTIVITY INTERRUPT SPDU
AIA	SPDU	ACTIVITY INTERRUPT ACK SPDU
AR	SPDU	ACTIVITY RESUME SPDU
AS	SPDU	ACTIVITY START SPDU
CD	SPDU	CAPABILITY DATA SPDU
CDA	SPDU	CAPABILITY DATA ACK SPDU
CDO	SPDU	CONNECT DATA OVERFLOW SPDU
CN	SPDU	CONNECT SPDU
DN	SPDU	DISCONNECT SPDU
DT	SPDU	DATA TRANSFER SPDU
ED	SPDU	EXCEPTION DATA SPDU
ER	SPDU	EXCEPTION REPORT SPDU
EX	SPDU	EXPEDITED DATA SPDU
FN-nr	SPDU	FINISH (not reuse) SPDU
FN-r	SPDU	FINISH (reuse) SPDU
GT	SPDU	GIVE TOKENS SPDU with Token Item parameter (Note 2)
GTA	SPDU	GIVE TOKENS ACK SPDU
GTC	SPDU	GIVE TOKENS CONFIRM SPDU
MAA	SPDU	MAJOR SYNC ACK SPDU
MAP	SPDU	MAJOR SYNC POINT SPDU
MIA	SPDU	MINOR SYNC ACK SPDU
MIP	SPDU	MINOR SYNC POINT SPDU
MIP-d	SPDU	MINOR SYNC POINT (data separation) SPDU
NF	SPDU	NOT FINISHED SPDU
OA	SPDU	OVERFLOW ACCEPT SPDU
PR-AB	SPDU	PREPARE (ABORT) SPDU
PR-MAA	SPDU	PREPARE (MAJOR SYNC ACK) SPDU
PR-RA	SPDU	PREPARE (RESYNCHRONIZE ACK) SPDU
PR-RS	SPDU	PREPARE (RESYNCHRONIZE) SPDU
PT	SPDU	PLEASE TOKENS SPDU with Token Item parameter (Notes 1 and 2)
RA	SPDU	RESYNCHRONIZE ACK SPDU
RF-nr	SPDU	REFUSE (not reuse) SPDU
RF-r	SPDU	REFUSE (reuse) SPDU
RS-a	SPDU	RESYNCHRONIZE (abandon) SPDU
RS-r	SPDU	RESYNCHRONIZE (restart) SPDU
RS-s	SPDU	RESYNCHRONIZE (set) SPDU
TD	SPDU	TYPED DATA SPDU

NOTES

1 If the ACCEPT SPDU contains the Token Item Parameter [see 7.4.1 c)], this shall be regarded by the SPM as if an ACCEPT SPDU had been received, followed by a PLEASE TOKENS SPDU. Therefore, both the AC event and the PT event occur.

2 GIVE TOKENS SPDU without Token Item parameter and PLEASE TOKENS SPDU without Token Item parameter are used to herald a concatenated sequence of SPDUs. Concatenation of SPDUs and separation of TSDUs are not handled by the state tables.

Table A.2 – States

Abbreviated name	Name and description
STA 01	Idle, no transport connection
STA 01A	Wait for the ABORT ACCEPT SPDU
STA 01B	Wait for T-CONNECT confirm
STA 01C	Idle, transport connected
STA 01D	Wait for the CONNECT DATA OVERFLOW SPDU
STA 02A	Wait for the ACCEPT SPDU
STA 02B	Wait for the OVERFLOW ACCEPT SPDU
STA 03	Wait for the DISCONNECT SPDU
STA 04A	Wait for the MAJOR SYNC ACK SPDU or the PREPARE (MAJOR SYNC ACK) SPDU
STA 04B	Wait for the ACTIVITY END ACK SPDU or the PREPARE (MAJOR SYNC ACK) SPDU
STA 05A	Wait for the RESYNCHRONIZE ACK SPDU or the PREPARE (RESYNCHRONIZE ACK) SPDU
STA 05B	Wait for the ACTIVITY INTERRUPT ACK SPDU or the PREPARE (RESYNCHRONIZE ACK) SPDU
STA 05C	Wait for the ACTIVITY DISCARD ACK SPDU or the PREPARE (RESYNCHRONIZE ACK) SPDU
STA 06	Wait for the RESYNCHRONIZE SPDU (resynchronization collision after receiving PREPARE (RESYNCHRONIZE) SPDU
STA 08	Wait for S-CONNECT response
STA 09	Wait for S-RELEASE response
STA 10A	Wait for S-SYNC-MAJOR response
STA 10B	Wait for S-ACTIVITY-END response
STA 11A	Wait for S-RESYNCHRONIZE response
STA 11B	Wait for S-ACTIVITY-INTERRUPT response
STA 11C	Wait for S-ACTIVITY-DISCARD response
STA 15A	After PREPARE, wait for the MAJOR SYNC ACK SPDU or the ACTIVITY END ACK SPDU
STA 15B	After PREPARE, wait for the RESYNCHRONIZE SPDU or the ACTIVITY INTERRUPT SPDU or the ACTIVITY DISCARD SPDU
STA 15C	After PREPARE, wait for the RESYNCHRONIZE ACK SPDU or the ACTIVITY INTERRUPT ACK SPDU or the ACTIVITY DISCARD ACK SPDU
STA 15D	After PREPARE, wait for the ABORT SPDU
STA 16	Wait for T-DISCONNECT indication
STA 18	Wait for the GIVE TOKENS ACK SPDU
STA 19	Wait for a recovery request or SPDU (initiator of EXCEPTION DATA SPDU)
STA 20	Wait for a recovery SPDU or request
STA 21	Wait for the CAPABILITY DATA ACK SPDU
STA 22	Wait for S-CAPABILITY-DATA response
STA 713	Data transfer state

Table A.3 – Outgoing events

Abbreviated name	Category	Name and description
SACTDind	SS-provider	S-ACTIVITY-DISCARD indication primitive
SACTDcnf	SS-provider	S-ACTIVITY-DISCARD confirm primitive
SACTEind	SS-provider	S-ACTIVITY-END indication primitive
SACTEcnf	SS-provider	S-ACTIVITY-END confirm primitive
SACTIind	SS-provider	S-ACTIVITY-INTERRUPT indication primitive
SACTIcnf	SS-provider	S-ACTIVITY-INTERRUPT confirm primitive
SACTRind	SS-provider	S-ACTIVITY-RESUME indication primitive
SACTSind	SS-provider	S-ACTIVITY-START indication primitive
SCDind	SS-provider	S-ACTIVITY-DATA indication primitive
SCDcnf	SS-provider	S-CAPABILITY-DATA confirm primitive
SCGind	SS-provider	S-CONTROL-GIVE indication primitive
SCONind	SS-provider	S-CONNECT indication primitive
SCONcnf+	SS-provider	S-CONNECT (accept) confirm primitive
SCONcnf–	SS-provider	S-CONNECT (reject) confirm primitive
SDTind	SS-provider	S-DATA indication primitive
SEXind	SS-provider	S-EXPEDITED-DATA indication primitive
SGTind	SS-provider	S-TOKEN-GIVE indication primitive
SPABind	SS-provider	S-P-ABORT indication primitive
SPERind	SS-provider	S-P-EXCEPTION-REPORT indication primitive
SPTind	SS-provider	S-TOKEN-PLEASE indication primitive
SRELind	SS-provider	S-RELEASE indication primitive
SRELcnf+	SS-provider	S-RELEASE (accept) confirm primitive
SRELcnf–	SS-provider	S-RELEASE (reject) confirm primitive
SRSYNind	SS-provider	S-RESYNCHRONIZE indication primitive
SRSYNcnf	SS-provider	S-RESYNCHRONIZE confirm primitive
SSYNMind	SS-provider	S-SYNC-MAJOR indication
SSYNMcnf	SS-provider	S-SYNC-MAJOR confirm
SSYNmind	SS-provider	S-SYNC-MINOR indication primitive
SSYNmdind	SS-provider	S-SYNC-MINOR (data separation) indication primitive
SSYNmcnf	SS-provider	S-SYNC-MINOR confirm primitive
STDind	SS-provider	S-TYPED-DATA indication primitive
SUABind	SS-provider	S-U-ABORT indication primitive
SUERind	SS-provider	S-U-EXCEPTION-REPORT indication primitive
TCONreq	TS-user	T-CONNECT request primitive
TCONrsp	TS-user	T-CONNECT response primitive
TDISreq	TS-user	T-DISCONNECT request primitive

Table A.3 *(concluded)* – Outgoing events

Abbreviated name	Category	Name and description
AA	SPDU	ABORT ACCEPT SPDU
AB-nr	SPDU	ABORT (not reuse) SPDU
AB-r	SPDU	ABORT (reuse) SPDU
AC	SPDU	ACCEPT SPDU
AD	SPDU	ACTIVITY DISCARD SPDU
ADA	SPDU	ACTIVITY DISCARD ACK SPDU
AE	SPDU	ACTIVITY END SPDU
AEA	SPDU	ACTIVITY END ACK SPDU
AI	SPDU	ACTIVITY INTERRUPT SPDU
AIA	SPDU	ACTIVITY INTERRUPT ACK SPDU
AR	SPDU	ACTIVITY RESUME SPDU
AS	SPDU	ACTIVITY START SPDU
CD	SPDU	CAPABILITY DATA SPDU
CDA	SPDU	CAPABILITY DATA ACK SPDU
CDO	SPDU	CONNECT DATA OVERFLOW SPDU
CN	SPDU	CONNECT SPDU
DN	SPDU	DISCONNECT SPDU
DT	SPDU	DATA TRANSFER SPDU
ED	SPDU	EXCEPTION DATA SPDU
EX	SPDU	EXPEDITED DATA SPDU
FN-nr	SPDU	FINISH (not reuse) SPDU
FN-r	SPDU	FINISH (reuse) SPDU
GT	SPDU	GIVE TOKENS SPDU
GTA	SPDU	GIVE TOKENS ACK SPDU
GTC	SPDU	GIVE TOKENS CONFIRM SPDU
MAA	SPDU	MAJOR SYNC ACK SPDU
MAP	SPDU	MAJOR SYNC POINT SPDU
MIA	SPDU	MINOR SYNC ACK SPDU
MIP	SPDU	MINOR SYNC POINT SPDU
MIP-d	SPDU	MINOR SYNC (data separation) POINT SPDU
NF	SPDU	NOT FINISHED SPDU
OA	SPDU	OVERFLOW ACCEPT SPDU
PR-AB	SPDU	PREPARE (ABORT) SPDU
PR-MAA	SPDU	PREPARE (MAJOR SYNC ACK) SPDU
PR-RA	SPDU	PREPARE (RESYNCHRONIZE ACK) SPDU
PR-RS	SPDU	PREPARE (RESYNCHRONIZE) SPDU
PT	SPDU	PLEASE TOKENS SPDU
RA	SPDU	RESYNCHRONIZE ACK SPDU
RF-nr	SPDU	REFUSE (not reuse) SPDU
RF-r	SPDU	REFUSE (reuse) SPDU
RS-a	SPDU	RESYNCHRONIZE (abandon) SPDU
RS-r	SPDU	RESYNCHRONIZE (restart) SPDU
RS-s	SPDU	RESYNCHRONIZE (set) SPDU
TD	SPDU	TYPED DATA SPDU

Table A.4 – Operations on variables when the symmetric synchronize functional unit has not been selected

Events	Condition for valid SPDU or primitive	Condition for update of variables	Operations on variables			
			V(A)	V(M)	V(R)	Vsc
SSYNMreq SSYNmreq SACTEreq		if Vsc true	set to V(M)	V(M) + 1	unchanged	false
		if Vsc false	unchanged	V(M) + 1	unchanged	false
MAP SPDU AE SPDU	sn = V(M)	if Vsc true	unchanged	V(M) + 1	unchanged	unchanged
		if Vsc false	set to V(M)	V(M) + 1	unchanged	unchanged
MIP SPDU	sn = V(M)	if Vsc true	unchanged	V(M) + 1	unchanged	true
		if Vsc false	set to V(M)	V(M) + 1	unchanged	true
SSYNMrsp SACTErsp MAA SPDU AEA SPDU	sn = V(M) – 1		set to V(M)	unchanged	set to V(M)	unchanged
SSYNmrsp	Vsc = true and V(M) > sn ≥ V(A)*		set to sn + 1	unchanged	unchanged	unchanged
MIA SPDU	Vsc = false and V(M) > sn ≥ V(A)*		set to sn + 1	unchanged	unchanged	unchanged
SRSYNreq	a: not applicable r: V(M) ≥ sn ≥ V(R) s: sn ≤ (10**ULSN) – 1	abandon restart set	unchanged unchanged unchanged	unchanged unchanged unchanged	unchanged unchanged unchanged	unchanged unchanged unchanged
RS SPDU	a: sn ≤ (10**ULSN) – 1 r: sn ≥ V(R) s: sn ≤ (10**ULSN) – 1	abandon restart set	unchanged unchanged unchanged	max sn, V(M) unchanged unchanged	unchanged unchanged unchanged	unchanged unchanged unchanged
SRSYNrsp	a: sn = V(M) r: sn as in RS SPDU s: sn ≤ (10**ULSN) – 1	abandon restart set	set to sn set to sn set to sn	set to sn set to sn set to sn	0 unchanged 0	unchanged unchanged unchanged
RA SPDU	a: sn ≥ V(M) r: sn as in RS SPDU s: sn ≤ (10**ULSN) – 1	abandon restart set	set to sn set to sn set to sn	set to sn set to sn set to sn	0 unchanged 0	unchanged unchanged unchanged
SACTRreq AR SPDU			set sn + 1	set sn + 1	set to 1	unchanged
SACTSreq AS SPDU			set to 1	set to 1	set to 1	unchanged
SCONrsp+ AC SPDU		sn present	set to sn	set to sn	0	false

sn	Synchronization point serial number quoted in SS-user request or SPDU
ULSN	Upper limit serial number negotiated during the connection establishment phase
≥	Greater than or equal to
≤	Less than or equal to
*	Synchronization point serial number not equal to V(M) – 1 if major synchronization or activity end outstanding

Table A.5 – Operations on variables when the symmetric synchronize functional unit has been selected

Events	Condition for valid SPDU or primitive	Condition for update of variables	Operations on variables					
			V(Ms)	V(Mr)	V(As)	V(Ar)	V(Rs)	V(Rr)
SSYNMreq SSYNmreq SACTEreq			V(Ms) + 1	unchanged	unchanged	unchanged	unchanged	unchanged
MAP SPDU AE SPDU	snr = V(Mr)		unchanged	V(Mr) + 1	unchanged	unchanged	unchanged	unchanged
MIP SPDU	snr = V(Mr)		unchanged	V(Mr) + 1	unchanged	unchanged	unchanged	unchanged
SSYNMrsp SACTErsp	sns = V(Ms) snr = V(Mr) – 1		V(Ms) + 1	unchanged	set to V(Ms)	set to V(Mr)	set to V(Ms)	set to V(Mr)
MAA SPDU AEA SPDU	sns = V(Ms) – 1 snr = V(Mr)		unchanged	V(Mr) + 1	set to V(Ms)	set to V(Mr)	set to V(Ms)	set to V(Mr)
SSYNmrsp	V(Mr) > snr ≥ V(Ar)		unchanged	unchanged	unchanged	set to snr + 1	unchanged	unchanged
MIA SPDU	V(Ms) > sns ≥ V(As)		unchanged	unchanged	set to sns + 1	unchanged	unchanged	unchanged
SRSYNreq *	r: V(Mr) ≥ snr ≥ V(Rr) V(Ms) ≥ sns ≥ V(Rs)		unchanged	unchanged	unchanged	unchanged	unchanged	unchanged
RS SPDU *	a: snr, sns r: snr ≥ V(Rr) sns ≥ V(Rs)		unchanged	unchanged	unchanged	unchanged	unchanged	unchanged
SRSYNrsp *	a: not applicable r: sn as in RS SPDU s: sn ≤ (10**ULSN) – 1	abandon restart set	set to sns set to sns set to sns	set to snr set to snr set to snr	set to sns set to sns set to sns	set to snr set to snr set to snr	0 unchanged 0	0 unchanged 0
RA SPDU *	a: snr ≥ V(Mr) sns ≥ V(Ms) r: sn as in RS SPDU s: sn ≤ (10**ULSN) – 1	abandon restart set	set to sns set to sns set to sns	set to snr set to snr set to snr	set to sns set to sns set to sns	set to snr set to snr set to snr	0 unchanged 0	0 unchanged 0
SACTRreq AR SPDU			set to sns + 1	set to snr + 1	set to sns + 1	set to snr + 1	set to 1	set to 1
SACTSreq AS SPDU			set to 1	set to 1	set to 1	set to 1	set to 1	set to 1
SCONrsp+ AC SPDU		snr and sns present	set to sns	set to snr	set to sns	set to snr	0	0

sns	Synchronization point serial number quoted in SS-user request or SPDU for the SPM sending flow
snr	Synchronization point serial number quoted in SS-user request or SPDU for the SPM receiving flow
ULSN	Upper limit serial number negotiated during the connection establishment phase
≥	Greater than or equal to
≤	Less than or equal to
*	If the Resync type is only specified for one direction of flow

Table A.6 – Specific actions

[1]	Set Vtca = true
[2]	Set Vtca = false
[3]	Stop timer TIM
[4]	Start timer TIM
[5]	Set V(A) = V(M) = serial number in ACCEPT SPDU Set V(R) = 0 Set Vcoll = false Set Vrsp = no Set Vsc = false Set V(Ado) = –1 Set V(Adi) = –1 Set TEXP Set FU(f) for f in fu-dom according to the intersection of Session User Requirements in the CONNECT SPDU and Session User Requirements in the ACCEPT SPDU If FU(ACT) = true, then set Vact = false Set Vdnr = false
[6]	Recall the queued events until the queue is empty
[7]	Set Vtrr = true
[8]	Set Vtrr = false
[9]	Set Vtrr according to the Transport Disconnect PV field in the SPDU As a local decision, Vtrr may always be set false
[10]	Store the event in the queue
[11]	Update the position of the tokens
[12]	Set Vact = true
[13]	Set Vnextact
[14]	Set Vact = Vnextact
[15]	Not used
[16]	If ¬FU(SS), then: Update Vrsp If RS-r, update Vrspnb Set Discard-rcv-flow = true Set Discard-snd-flow = true If FU(SS), then: Update Vrsps If Vrsps = r, update Vrspnbs If Vrsps ≠ no, set Discard-snd-flow = true If FU(SS), then: Update Vrspr If Vrspr = r, update Vrspnbr If Vrspr ≠ no, set Discard-rcv-flow = true
[17]	Not used
[18]	Set Vcoll = true
[19]	If ¬FU(SS), then: V(M) = maximum [V(M), received serial number] If FU(SS), then: V(Mr) = maximum [V(Mr), received first serial number] V(Ms) = maximum [V(Ms), received second serial number]
[20]	Set Vsc = false
[21]	Set V(M) = V(M) + 1
[22]	Set V(R) = V(A) = V(M)
[23]	If Vsc = false, then set V(A) = V(M) Set Vsc = true Set V(M) = V(M) + 1
[24]	If Vsc = true, then set V(A) = V(M) Set Vsc = false Set V(M) = V(M) + 1
[25]	Set V(A) = serial number + 1
[26]	Set V(A) = V(M) = V(R) = 1
[27]	Set V(A) = V(M) = serial number + 1 Set V(R) = 1

Table A.6 *(concluded)* – Specific actions

[28]	Set V(A) = V(M) = serial number If Vrsp = a, then set V(R) = 0 If Vrsp = s, then set V(R) = 0 Set Vrsp = no Set V(Ado) = –1 Set V(Adi) = –1 Set Discard-rcv-flow = false Set Discard-snd-flow = false
[29]	Set the position of the tokens such that all available tokens are owned Set Vact = false Set Vrsp = no
[30]	Set the position of the tokens such that all available tokens are not owned Set Vact = false Set Vrsp = no
[31]	If Vsc = false, then set V(A) = V(M) Set V(M) = V(M) + 1
[32]	Set Vdnr = true
[41]	Set V(Ado) = V(M)
[42]	Set V(Ado) = V(Ms)
[43]	Set V(Ado) = –1 Set V(Adi) = –1
[44]	Set V(Adi) = V(M)
[45]	Set V(Adi) = V(Mr)
[50]	Preserve user data for subsequent SCONind
[51]	If p201 send subsequent CDO SPDUs until ¬p201
[62]	Set V(Mr) = V(Mr) + 1
[63]	Set V(Ms) = V(Ms) + 1
[64]	Set V(Rs) = V(As) = V(Ms) Set V(Rr) = V(Ar) = V(Mr)
[65]	Set V(As) = serial number + 1
[66]	Set V(Ar) = serial number + 1
[67]	If neither Resync Type is present, then: Set Discard-rcv-flow = true Set Discard-snd-flow = true If Resync Type is present for sending flow, then: Set Discard-snd-flow = true If Resync Type is present for receiving flow, then: Set Discard-rcv-flow = true
[68]	Set V(As) = V(Ms) = sending flow serial number, if present If Vrsps = a, then set V(Rs) = 0 If Vrsps = s, then set V(Rs) = 0 Set Vrsps = no Set Discard-snd-flow = false Set V(Ar) = V(Mr) = receiving flow serial number, if present If Vrspr = a, then set V(Rr) = 0 If Vrspr = s, then set V(Rr) = 0 Set Vrspr = no Set Discard-rcv-flow = false Set V(Ado) = –1 Set V(Adi) = –1
[69]	Set V(As) = V(Ms) = sending flow serial number + 1 Set V(Rs) = 1 Set V(Ar) = V(Mr) = receiving flow serial number + 1 Set V(Rr) = 1
[70]	Set V(As) = V(Ms) = V(Rs) = 1 Set V(Ar) = V(Mr) = V(Rr) = 1
[71]	Set V(As) = V(Ms) = sending flow serial number in ACCEPT SPDU Set V(Rs) = 0 Set Vrsps = no Set Discard-snd-flow = false Set V(Ar) = V(Mr) = receiving flow serial number in ACCEPT SPDU Set V(Rr) = 0 Set Vrspr = no Set Discard-rcv-flow = false Set Vcoll = false Set V(Ado) = –1 Set V(Adi) = –1
[72]	If FU(SS), then update Vrspr, Vrsps, Vrspnbr, and Vrspnbs as in A.5.4.5

Table A.7 – Predicates

p01	¬Vtca
p02	local choice & ¬TEXP
p03	I(dk)
p04	FU(FD) & ¬Vcoll
p05	A(dk)
p06	FU(TD)
p07	FU(TD) & ¬Vcoll
p08	FU(EX)
p09	FU(EX) & ¬Vcoll
p10	¬Vcoll
p11	II(ma)
p12	FU(MA) & [¬FU(ACT) OR Vact] & A(dk) & A(mi) & AA(ma)
p13	FU(MA) & [¬FU(ACT) OR Vact] & I(dk) & I(mi) & II(ma)
p14	[¬FU(ACT) OR Vact] & A(dk) & AA(mi)
p15	[¬FU(ACT) OR Vact] & I(dk) & II(mi)
p16	¬TEXP
p17	[¬FU(ACT) OR Vact] & [FU(SS) OR FU(SY)] & [FU(SS) OR ¬Vsc]
p18	[¬FU(ACT) OR Vact] & [FU(SS) OR FU(SY)] & [FU(SS) OR Vsc]
p19	¬FU(SS) & [serial number = V(M)]
p20	¬FU(SS) & [serial number = V(M) – 1]
p21	¬FU(SS) & [V(M) > serial number ≥ V(A)]
p22	Unused
p23	FU(ACT) & ¬Vnextact
p24	¬SPMwinner
p25	[FU(SY) OR FU(SS) OR FU(MA)] & FU(RESYN)
p26	[¬FU(ACT) OR Vact]
p27	Vrsp = no
p28	FU(RESYN)
p29	[¬FU(ACT) OR Vact] & FU(RESYN)
p30	¬FU(ACT) OR Vnextact
p31	FU(ACT) & Vnextact
p32	[¬FU(SS) & [[type ≠ r] OR [serial number ≥ V(R)]]] OR [FU(SS) & [[rcv flow type ≠ r] OR [rcv flow serial number ≥ V(Rr)]] & [[snd flow type ≠ r] OR [snd flow serial number ≥ V(Rs)]]]
p33	V(M) ≥ serial number ≥ V(R)
p34	FU(ACT)
p35	FU(RESYN) & [¬TEXP OR FU(DS)]
p36	FU(RESYN) & TEXP
p37	FU(ACT) & TEXP
p38	FU(ACT) & ¬TEXP
p39	Vact & II(ma)
p40	AA(ma)
p41	Vrsp = dsc
p42	Vrsp = int
p43	¬FU(SS) & [[[Vrsp = r] & [serial number = Vrspnb]] OR [[Vrsp = a] & [serial number = V(M)]] OR [Vrsp = s]]
p44	[FU(ACT) & ¬Vact] & A(dk) & A(mi) & A(ma)
p45	[FU(ACT) & ¬Vact] & I(dk) & I(mi) & I(ma)
p46	FU(CD) & [FU(ACT) & ¬Vact] & A(dk) & A(mi) & ¬OWNED(ma)
p47	FU(CD) & [FU(ACT) & ¬Vact] & I(dk) & I(mi) & OWNED(ma)
p48	FU(EXCEP) & FU(HD)

Table A.7 *(concluded)* – Predicates

p49	[[Vrsp = r] & [serial number = Vrspnb]] OR [[Vrsp = a] & [serial number ≥ V(M)]] OR [Vrsp = s]
p50	FU(EXCEP) & [¬FU(ACT) OR Vact] & AA(dk)
p51	FU(EXCEP) & [¬FU(ACT) OR Vact] & II(dk)
p52	FU(EXCEP) & ¬FU(ACT) & II(dk)
p53	ALL(AV, RT)
p54	ALL(II, GT)
p55	[FU(ACT) & ¬Vact] & ALL(I, tk-dom)
p57	ALL(II, GT) & (dk not in GT)
p58	ALL(II, GT) & (dk in GT)
p59	ALL(AA, GT)
p60	ALL(AA, GT) & (dk not in GT)
p61	ALL(AA, GT) & (dk in GT)
p62	[FU(ACT) & ¬Vact] & ALL(A, tk-dom)
p63	ALL(I, tk-dom) & [¬FU(ACT) OR ¬Vact]
p64	local choice & ¬Vtca & ¬TEXP
p65	ANY(AV, tk-dom)
p66	Vtrr
p67	FU(NR)
p68	ALL(A, tk-dom) & [¬FU(ACT) OR ¬Vact]
p69	Vcoll
p70	FU(FD)
p71	FU(ACT) & Vact & I(dk) & I(mi) & II(ma)
p72	FU(ACT) & Vact & A(dk) & A(mi) & AA(ma)
p75	(Vcoll & Vdnr) OR ¬Vcoll
p76	CN SPDU is not acceptable to the SPM for transient or persistent reason (see 8.3.5.10)
p80	¬FU(DS) OR [¬FU(SS) & V(Adi) < V(A)] OR [FU(SS) & V(Adi) < V(Ar)]
p81	[¬FU(SS) & V(Ado) ≥ V(A)] OR [FU(SS) & V(Ado) ≥ V(As)]
p82	FU(DS)
p173	FU(SS) & [serial numbers = V(Ms) – 1, V(Mr)]
p174	FU(SS) & [serial numbers = V(Ms), V(Mr) – 1]
p175	FU(SS) & [serial number = V(Mr) – 1]
p176	FU(SS) & [V(Ms) > serial number ≥ V(As)]
p177	FU(SS) & [serial number = V(Mr)]
p178	FU(SS)
p179	FU(SS) & [V(Mr) > serial number ≥ V(Ar)]
p180	[¬FU(SS) & [type ≠ r]] OR [FU(SS) & [[rcv flow type = a] OR [rcv flow type = s]] & [[snd flow type = a] OR [snd flow type = s]]]
p184	FU(SS) & Vrsps = no
p185	Discard-rcv-flow & ¬p81
p186	Discard-snd-flow
p187	FU(SS) & [[Vrspr ≠ r] OR [rcv flow serial number = Vrspnbr]] & [[Vrsps ≠ r] OR [snd flow serial number = Vrspnbs]]
p201	More user data to send
p202	End of user data
p204	More than 10 240 octets of SS-user data to be transferred

Table A.8 – Connection establishment state table without the symmetric synchronize functional unit

State / Event	STA01 idle no TC	STA01A await AA	STA01B await TCONcnf	STA01C idle TC con	STA01D await CDO	STA02A await AC	STA02B await OA	STA08 await SCONrsp	STA15D wait after PR-AB	STA16 await TDISind
AC	//	STA01A	//	TDISreq STA01		SCONcnf+ [5][11] STA713 [6]			STA15D	STA16
CDO	//		//	TDISreq STA01	¬p202 [50] STA01D p202 SCONind STA08				STA15D	
CN	//	TDISreq [3] STA01	//	¬p01&¬p76&p204 OA [50] STA01D ¬p01&¬p76&¬p204 SCONind STA08 ¬p01&p76&¬p02 RF-nr [4] STA16 ¬p01&p76&p02 RF-r STA01C p01 TDISreq STA01						TDISreq [3] STA01
OA	//		//	TDISreq STA01			CDO [51] STA02A		STA15D	
RF-nr	//	STA01A	//	TDISreq STA01		SCONcnf– TDISreq STA01	SCONcnf– TDISreq STA01			STA16
RF-r	//	STA01A	//	TDISreq STA01		¬p02 SCONcnf– TDISreq STA01 p02 SCONcnf– STA01C	¬p02 SCONcnf– TDISreq STA01 p02 SCONcnf– STA01C			STA16
SCONreq	TCONreq [2] STA01B			p01&p204 CN STA02B p01&¬p204 CN STA02A						
SCONrsp+								AC [5][11] STA713	STA15D	
SCONrsp–								¬p02 RF-nr [4] STA16 p02 RF-r STA01C	[4] STA16	
TCONcnf	//	//	p204 CN STA02B ¬p204 CN STA02A	//	//	//	//	//	//	//
TCONind	TCONrsp [1] STA01C	//	//	//	//	//	//	//	//	//

Table A.9 – Data transfer state table without the symmetric synchronize functional unit

State / Event	STA01A await AA	STA01C idle TC con	STA01D await CDO	STA02A await AC	STA03 await DN	STA04A await PR or MAA	STA04B await PR or AEA	STA05A await PR or RA	STA05B await PR or AIA
DT	STA01A	TDISreq STA01	TDISreq STA01		p05&p10 SDTind STA03	p05 SDTind STA04A	p05 SDTind STA04B	p05&p185 STA05A p05&¬p185 SDTind STA05A	p05 STA05B
EX	STA01A	TDISreq STA01	TDISreq STA01	[10] STA02A	p09 SEXind STA03	p08 SEXind STA04A	p08 SEXind STA04B	p08&p185 STA05A p08&¬p185 SEXind STA05A	p08 STA05B
TD	STA01A	TDISreq STA01	TDISreq STA01		p06&p10 STDind STA03	p06 STDind STA04A	p06 STDind STA04B	p06&p185 STA05A p06&¬p185 STDind STA05A	p06 STA05B
SDTreq									
SEXreq									
STDreq									

Table A.9 *(continued)* – Data transfer state table without the symmetric synchronize functional unit

State / Event	STA05C await PR or ADA	STA06 await RS after coll	STA09 await SRELrsp	STA10A await SSYNMrsp	STA10B await SACTErsp	STA11A await SRSYNrsp	STA15A wait after PR-MAA	STA15B wait after PR-RS
DT	p05 STA05C	p05&p185 STA06 p05&¬p185 SDTind STA06				p05&¬p185 SDTind STA11A	p05 SDTind STA15A	p05&p185 STA15B p05&¬p185 SDTind STA15B
EX	p08 STA05C	p08 [10] STA06				p08&¬p185 SEXind STA11A	p08 [10] STA15A	
TD	p06 STA05C	p06&p185 STA06 p06&¬p185 STDind STA06					p06 STDind STA15A	p06&p185 STA15B p06&¬p185 STDind STA15B
SDTreq			p04 DT STA09	p03 DT STA10A	p03 DT STA10B			p03 STA15B
SEXreq			p09 EX STA09	p08 EX STA10A	p08 EX STA10B			p08 STA15B
STDreq			p07 TD STA09	p06 TD STA10A	p06 TD STA10B			p06 STA15B

Table A.9 *(concluded)* – Data transfer state table without the symmetric synchronize functional unit

State / Event	STA15C wait after PR-RA	STA15D wait after PR-AB	STA16 await TDISind	STA18 await GTA	STA19 await recovery(init)	STA20 await recovery	STA21 await CDA	STA713 data transfer
DT	p05&p185 STA15C p05&¬p185 SDTind STA15C	STA15D	STA16	p70 SDTind STA18	STA19	p05 STA20	p70 SDTind STA21	p05 SDTind STA713
EX	p08 [10] STA15C		STA16	p08 SEXind STA18	p08 STA19	p08 STA20	p08 SEXind STA21	p08 SEXind STA713
TD	p06&p185 STA15C p06&¬p185 STDind STA15C	STA15D	STA16	p06 STDind STA18	p06 STA19	p06 STA20	p06 STDind STA21	p06 STDind STA713
SDTreq		STA15D		p70 DT STA18				p03 DT STA713
SEXreq		STA15D		p08 EX STA18				p08 EX STA713
STDreq		STA15D		p06 TD STA18				p06 TD STA713

Table A.10 – Synchronization state table without the symmetric synchronize functional unit

State / Event	STA01A await AA	STA01C idle TC con	STA01D await CDO	STA04A await PR or MAA	STA04B await PR or AEA	STA05A await PR or RA	STA05B await PR or AIA	STA05C await PR or ADA	STA06 await RS after coll
MAA/AEA	STA01A	TDISreq STA01	TDISreq STA01	p16&p20 SSYNMcnf [14][22] STA713	p16&p20 SACTEcnf [14][22] STA713	[43] STA05A	[43] STA05B	[43] STA05C	[43] STA06
MAP	STA01A	TDISreq STA01	TDISreq STA01			p12&p185 STA05A p12&¬p178&¬p185 SSYNMind [23] STA05A			p12&p185 STA06 p12&¬p178&¬p185 SSYNMind [23] STA06
PR-MAA	STA01A	TDISreq STA01	TDISreq STA01	STA15A	STA15A	STA05A	STA05B	STA05C	
SSYNMreq									
SSYNMrsp									

Table A.10 *(continued)* – Synchronization state table without the symmetric synchronize functional unit

State / Event	STA10A await SSYNMrsp	STA15A wait after PR-MAA	STA15B wait after PR-RS	STA15C wait after PR-RA	STA15D wait after PR-AB	STA16 await TDISind	STA19 await recovery(init)	STA20 await recovery	STA713 data transfer
MAA/AEA		p20&¬p23 SSYNMcnf [14][22] STA713 [6] p20&p23 SACTEcnf [14][22] STA713 [6]	[43] STA15B	[43] STA15C	STA15D	STA16		p20 STA20	
MAP			p12 STA15B	p12&p185 STA15C p12&¬p178&¬p185 SSYNMind [23] STA15C	STA15D	STA16	p12&p19 [31] STA19	p12&p19 [31] STA20	p12&p19 SSYNMind [13][31] STA10A
PR-MAA						STA16			
SSYNMreq			p13 STA15B		STA15D				p13 MAP [13][24] STA04A
SSYNMrsp	PR-MAA(1) MAA [14][22] STA713		STA15B		STA15D				

Table A.10 *(continued)* – Synchronization state table without the symmetric synchronize functional unit

State / Event	STA01A await AA	STA01C idle TC con	STA01D await CDO	STA03 await DN	STA04A await PR or MAA	STA04B await PR or AEA	STA05A await PR or RA	STA05B await PR or AIA
AE	STA01A	TDISreq STA01	TDISreq STA01				p72 STA05A	
MIA	STA01A	TDISreq STA01	TDISreq STA01	p17&p21 SSYNmcnf [25] STA03	p17&¬p20&p21 SSYNmcnf [25] STA04A	p17&¬p20&p21 SSYNmcnf [25] STA04B	p17&¬p81 STA05A p17&p21&p81 SSYNmcnf [25] STA05A	p17 STA05B
MIP	STA01A	TDISreq STA01	TDISreq STA01				p14&¬p178&p185 STA05A ¬p178&¬p185&p26 SSYNmind [23] STA05A	p14 STA05B
MIP-d	STA01A	TDISreq STA01	TDISreq STA01				p14&p82 [44] STA05A p82&¬p178&¬p185&p26 SSYNmind [44][23] STA05A	p14&p82 [44] STA05B
SACTEreq								
SACTErsp								
SSYNmdreq								
SSYNmreq								
SSYNmrsp								

Table A.10 *(continued)* – Synchronization state table without the symmetric synchronize functional unit

State / Event	STA05C await PR or ADA	STA06 await RS after coll	STA09 await SRELrsp	STA10A await SSYNMrsp	STA10B await SACTErsp	STA15A wait after PR-MAA
AE		p72 STA06				
MIA	p17 STA05C	p17&¬p81 STA06 p17&p21&p81 SSYNmcnf [25] STA06				p17&¬p20&p21 SSYNmcnf [25] STA15A
MIP	p14 STA05C	p14 STA06 ¬p178&¬p185&p26 SSYNmind [23] STA06				
MIP-d	p14&p82 [44] STA05C	p14&p82 [44] STA06 p82&¬p178&¬p185&p26 SSYNmind [44][23] STA06				
SACTEreq						
SACTErsp					PR-MAA(1) AEA [14][22] STA713	
SSYNmdreq						
SSYNmreq						
SSYNmrsp			p18&p21 MIA [25] STA09	p18&¬p20&p21 MIA [25] STA10A	p18&¬p20&p21 MIA [25] STA10B	

Table A.10 *(concluded)* – Synchronization state table without the symmetric synchronize functional unit

State / Event	STA15B wait after PR-RS	STA15C wait after PR-RA	STA15D wait after PR-AB	STA16 await TDISind	STA19 await recovery(init)	STA20 await recovery	STA713 data transfer
AE	p72 STA15B	p72 STA15C	STA15D	STA16	p72&p19 [31] STA19	p72&p19 [31] STA20	p72&p19 SACTEind [13][31] STA10B
MIA	p17&¬p81 STA15B p17&p21&p81 SSYNmcnf [25] STA15B	p17&¬p81 STA15C p17&p21&p81 SSYNmcnf [25] STA15C	STA15D	STA16	p17&p21 [25] STA19	p17&p21 STA20	p17&p21 SSYNmcnf [25] STA713
MIP	p14&¬p178&p185 STA15B p14&¬p178&¬p185 SSYNmind [23] STA15B	p14&¬p178&p185 STA15C p14&¬p178&¬p185 SSYNmind [23] STA15C	STA15D	STA16	p14&p19 [23] STA19	p14&p19 [23] STA20	p14&p19 SSYNmind [23] STA713
MIP-d		p14&p82&¬p178&p185 [44] STA15C p14&p82&¬p178&¬p185 SSYNmind [44][23] STA15C	p82 [44] STA15D	p82 [44] STA16	p14&p19&p82 [44][23] STA19	p14&p19&p82 [44][23] STA20	p14&p19&p82 SSYNmind [44][23] STA713
SACTEreq	p71 STA15B		STA15D				p71 AE [13][24] STA04B
SACTErsp			STA15D				
SSYNmdreq	p15&p82 STA15B		STA15D				p15&p82 MIP [41][24] STA713
SSYNmreq	p15 STA15B		STA15D				p15 MIP [24] STA713
SSYNmrsp	p18&p21 STA15B		STA15D				p18&p21 MIA [25] STA713

Table A.11 – Resynchronization state table without the symmetric synchronize functional unit

State / Event	STA01A await AA	STA01C idle TC con	STA01D await CDO	STA02A await AC	STA03 await DN	STA04A await PR or MAA	STA04B await PR or AEA	STA05A await PR or RA
PR-RA	STA01A	TDISreq STA01	TDISreq STA01					STA15C
PR-RS	STA01A	TDISreq STA01	TDISreq STA01	[10] STA02A	p10&p80 STA15B	p80 STA15B	p80 STA15B	p80 STA06
RA	STA01A	TDISreq STA01	TDISreq STA01					p35&p49 SRSYNcnf [28][11] STA713
RS-a	STA01A	TDISreq STA01	TDISreq STA01		p10&¬p34&p35 [19] SRSYNind(2) [16] STA11A	p35 [19] SRSYNind(2) [16] STA11A	p35 [19] SRSYNind(2) [16] STA11A	¬p24&p35 STA05A p24&p35 [19] SRSYNind(2) [16] STA11A
RS-r	STA01A	TDISreq STA01	TDISreq STA01		p10&¬p34&p35&p32 SRSYNind [16] STA11A	p32&p35 SRSYNind [16] STA11A	p32&p35 SRSYNind [16] STA11A	¬p24&p32&p35 STA05A p24&p32&p35 SRSYNind [16] STA11A
RS-s	STA01A	TDISreq STA01	TDISreq STA01		p10&¬p34&p35 SRSYNind [16] STA11A	p35 SRSYNind [16] STA11A	p35 SRSYNind [16] STA11A	¬p24&p35 STA05A p24&p35 SRSYNind [16] STA11A
SRSYNreq(a)						p28 PR-RS(5) RS-a [16] STA05A		
SRSYNreq(r)								
SRSYNreq(s)						p28 PR-RS(5) RS-s [16] STA05A		
SRSYNrsp								

Table A.11 *(continued)* – Resynchronization state table without the symmetric synchronize functional unit

State / Event	STA05B await PR or AIA	STA05C await PR or ADA	STA06 await RS after coll	STA09 await SRELrsp	STA10A await SSYNMrsp	STA10B await SACTErsp	STA11A await SRSYNrsp	STA15A wait after PR-MAA
PR-RA	STA15C	STA15C	[10] STA06					
PR-RS	STA05B	STA05C	[10] STA06		p80 STA15B	p80 STA15B		[10] STA15A
RA								
RS-a	p28 STA05B	p28 STA05C	¬p24 STA05A [6] p24 [19] SRSYNind(2) [16] STA11A [6]		p35 [19] SRSYNind(2) [16] STA11A			
RS-r	p28 STA05B	p28 STA05C	¬p24&p32 STA05A [6] p24&p32 SRSYNind [16] STA11A [6]					
RS-s	p28 STA05B	p28 STA05C	¬p24 STA05A [6] p24 SRSYNind [16] STA11A [6]		p35 SRSYNind [16] STA11A			
SRSYNreq(a)				p10&p28&¬p34 PR-RS(5) RS-a [16] STA05A	p28 PR-RS(5) RS-a [16] STA05A	p28 PR-RS(5) RS-a [16] STA05A	p24 PR-RS(5) RS-a [16] STA05A	p28&p30 PR-RS(5) RS-a [16] STA05A [6]
SRSYNreq(r)				p10&p25&¬p34&p33 PR-RS(5) RS-r [16] STA05A	p25&p33 PR-RS(5) RS-r [16] STA05A	p25&p33 PR-RS(5) RS-r [16] STA05A	p24&p33 PR-RS(5) RS-r [16] STA05A	
SRSYNreq(s)				p10&p25&¬p34 PR-RS(5) RS-s [16] STA05A	p25 PR-RS(5) RS-s [16] STA05A	p25 PR-RS(5) RS-s [16] STA05A	p24 PR-RS(5) RS-s [16] STA05A	p28&p30 PR-RS(5) RS-s [16] STA05A [6]
SRSYNrsp							p43 PR-RA(1) RA [28][11] STA713	

Table A.11 *(concluded)* – Resynchronization state table without the symmetric synchronize functional unit

State / Event	STA15B wait after PR-RS	STA15C wait after PR-RA	STA15D wait after PR-AB	STA16 await TDISind	STA18 await GTA	STA19 await recovery(init)	STA20 await recovery	STA713 data transfer
PR-RA				STA16				
PR-RS		[10] STA15C		STA16	[10] STA18	p80 STA15B	p80 STA15B	p26 STA15B ¬p26 [10] STA713
RA		p36&p49 SRSYNcnf [28][11] STA713 [6]	STA15D	STA16				
RS-a	p29 [19] SRSYNind(2) [16] STA11A		STA15D	STA16		p35 [19] SRSYNind(2) [16] STA11A	p35 [19] SRSYNind(2) [16] STA11A	p26&p35 [19] SRSYNind(2) [16] STA11A
RS-r	p32&p29 SRSYNind [16] STA11A		STA15D	STA16		p32&p35 SRSYNind [16] STA11A	p32&p35 SRSYNind [16] STA11A	p32&p26&p35 SRSYNind [16] STA11A
RS-s	p29 SRSYNind [16] STA11A		STA15D	STA16		p35 SRSYNind [16] STA11A	p35 SRSYNind [16] STA11A	p26&p35 SRSYNind [16] STA11A
SRSYNreq(a)	p27&p28 PR-RS(5) RS-a [16] STA06		STA15D				p28 PR-RS(5) RS-a [16] STA05A	p29 PR-RS(5) RS-a [16] STA05A
SRSYNreq(r)	p25&p27&p33 PR-RS(5) RS-r [16] STA06		STA15D				p25&p33 PR-RS(5) RS-r [16] STA05A	p25&p26&p33 PR-RS(5) RS-r [16] STA05A
SRSYNreq(s)	p25&p27 PR-RS(5) RS-s [16] STA06		STA15D				p25 PR-RS(5) RS-s [16] STA05A	p25&p26 PR-RS(5) RS-s [16] STA05A
SRSYNrsp			STA15D					

Table A.12 – Activity interrupt and discard state table without the symmetric synchronize functional unit

State / Event	STA01A await AA	STA01C idle TC con	STA01D await CDO	STA04A await PR or MAA	STA04B await PR or AEA	STA05A await PR or RA	STA05B await PR or AIA	STA05C await PR or ADA	STA06 await RS after coll
AD	STA01A	TDISreq STA01	TDISreq STA01			p38&p40 SACTDind [16] STA11C			p37&p40 SACTDind [16] STA11C
ADA	STA01A	TDISreq STA01	TDISreq STA01					p38 SACTDcnf [29] STA713	
AI	STA01A	TDISreq STA01	TDISreq STA01			p38&p40 SACTIind [16] STA11B			p37&p40 SACTIind [16] STA11B
AIA	STA01A	TDISreq STA01	TDISreq STA01				p38 SACTIcnf [29] STA713		
SACTDreq				p34&p39 PR-RS(1) AD [16] STA05C	p39 PR-RS(1) AD [16] STA05C				
SACTDrsp									
SACTIreq				p34&p39 PR-RS(1) AI [16] STA05B	p39 PR-RS(1) AI [16] STA05B				
SACTIrsp									

Table A.12 *(continued)* – Activity interrupt and discard state table without the symmetric synchronize functional unit

State / Event	STA10A await SSYNMrsp	STA10B await SACTErsp	STA11A await SRSYNrsp	STA11B await SACTIrsp	STA11C await SACTDrsp	STA15A wait after PR-MAA	STA15B wait after PR-RS
AD	p38&p40 SACTDind [16] STA11C	p38&p40 SACTDind [16] STA11C					p37&p40 SACTDind [16] STA11C
ADA							
AI	p38&p40 SACTIind [16] STA11B	p38&p40 SACTIind [16] STA11B					p37&p40 SACTIind [16] STA11B
AIA							
SACTDreq			p34&p39 PR-RS(1) AD [16] STA05C			p34&p39 PR-RS(1) AD [16] STA05C [6]	p27&p34&p39 PR-RS(1) AD [16] STA05C
SACTDrsp					PR-RA(1) ADA [30] STA713		
SACTIreq			p34&p39 PR-RS(1) AI [16] STA05B			p34&p39 PR-RS(1) AI [16] STA05B [6]	p27&p34&p39 PR-RS(1) AI [16] STA05B
SACTIrsp				PR-RA(1) AIA [30] STA713			

Table A.12 *(concluded)* – Activity interrupt and discard state table without the symmetric synchronize functional unit

State / Event	STA15C wait after PR-RA	STA15D wait after PR-AB	STA16 await TDISind	STA19 await recovery(init)	STA20 await recovery	STA21 await CDA	STA22 await SCDrsp	STA713 data transfer
AD		STA15D	STA16	p38&p40 SACTDind [16] STA11C	p38&p40 SACTDind [16] STA11C			p38&p40 SACTDind [16] STA11C
ADA	p37&p41 SACTDcnf [29] STA713 [6]	STA15D	STA16					
AI		STA15D	STA16	p38&p40 SACTIind [16] STA11B	p38&p40 SACTIind [16] STA11B		p16 SACTIind [16] STA11B	p38&p40 SACTIind [16] STA11B
AIA	p37&p42 SACTIcnf [29] STA713 [6]	STA15D	STA16					
SACTDreq		STA15D			p34&p11 PR-RS(1) AD [16] STA05C			p34&p39 PR-RS(1) AD [16] STA05C
SACTDrsp		STA15D						
SACTIreq		STA15D			p34&p11 PR-RS(1) AI [16] STA05B	p16 AI [16] STA05B		p34&p39 PR-RS(1) AI [16] STA05B
SACTIrsp		STA15D						

Table A.13 – Activity start, resume and capability data state table without the symmetric synchronize functional unit

State / Event	STA01A await AA	STA01C idle TC con	STA01D await CDO	STA05B await PR or AIA	STA15B wait after PR-RS	STA15D wait after PR-AB	STA16 await TDISind	STA21 await CDA	STA22 await SCDrsp	STA713 data transfer
AR	STA01A	TDISreq STA01	TDISreq STA01		p44 SACTRind [12][27] STA15B	STA15D	STA16			p44 SACTRind [12][27] STA713 [6]
AS	STA01A	TDISreq STA01	TDISreq STA01		p44 SACTSind [12][26] STA15B	STA15D	STA16			p44 SACTSind [12][26] STA713 [6]
CD	STA01A	TDISreq STA01	TDISreq STA01			STA15D	STA16			p46 SCDind STA22
CDA	STA01A	TDISreq STA01	TDISreq STA01	¬p26 STA05B		STA15D	STA16	SCDcnf STA713		
SACTRreq						STA15D				p45 AR [12][27] STA713
SACTSreq										p45 AS [12][26] STA713
SCDreq						STA15D				p47 CD STA21
SCDrsp						STA15D			CDA STA713	

Table A.14 – Token management and exceptions state table without the symmetric synchronize functional unit

State / Event	STA01A await AA	STA01C idle TC con	STA01D await CDO	STA03 await DN	STA04A await PR or MAA	STA04B await PR or AEA	STA05A await PR or RA	STA05B await PR or AIA
ED	STA01A	TDISreq STA01	TDISreq STA01	p52 SUERind STA20	p48&p03 SUERind STA20 p48&¬p03 SUERind STA713	p48&p03 SUERind STA20 p48&¬p03 SUERind STA713	p48 STA05A	p48 STA05B
ER	STA01A	TDISreq STA01	TDISreq STA01	p52 SPERind STA20	p48&p03 SPERind STA20 p48&¬p03 SPERind STA713	p48&p03 SPERind STA20 p48&¬p03 SPERind STA713	p48 STA05A	p48 STA05B
GT	STA01A	TDISreq STA01	TDISreq STA01		p59 SGTind [11] STA04A	p59 SGTind [11] STA04B	p59&p185 STA05A p59&¬p185 SGTind [11] STA05A	p59 STA05B
GTA	STA01A	TDISreq STA01	TDISreq STA01					
GTC	STA01A	TDISreq STA01	TDISreq STA01					
PT	STA01A	TDISreq STA01	TDISreq STA01	p53 SPTind STA03	p53 SPTind STA04A	p53 SPTind STA04B	p53&p185 STA05A p53&¬p185 SPTind STA05A	p53 STA05B
SCGreq								
SGTreq					p54 GT [11] STA04A	p54 GT [11] STA04B		
SPTreq								
SUERreq								

Table A.14 *(continued)* – Token management and exceptions state table without the symmetric synchronize functional unit

State / Event	STA05C await PR or ADA	STA06 await RS after coll	STA09 await SRELrsp	STA10A await SSYNMrsp	STA10B await SACTErsp	STA15A wait after PR-MAA	STA15B wait after PR-RS
ED	p48 STA05C	p48 STA06					
ER	p48 STA05C	p48 STA06					
GT	p59 STA05C	p59&p185 STA06 p59&¬p185 SGTind [11] STA06		p59 SGTind [11] STA10A	p59 SGTind [11] STA10B	p59 SGTind [11] STA15A	p59&p185 STA15B p59&¬p185 SGTind [11] STA15B
GTA							
GTC							
PT	p53 STA05C	p53&p185 STA06 p53&¬p185 SPTind STA06				p53 SPTind STA15A	p53&p185 STA15B p53&¬p185 SPTind STA15B
SCGreq							
SGTreq				p54 GT [11] STA10A	p54 GT [11] STA10B	p54 GT [11] STA15A	p54 STA15B
SPTreq			p53 PT STA09	p53 PT STA10A	p53 PT STA10B		p53 STA15B
SUERreq			p50 ED STA19	p50 ED STA19	p50 ED STA19		p50 STA15B

Table A.14 *(concluded)* – Token management and exceptions state table without the symmetric synchronize functional unit

State / Event	STA15C wait after PR-RA	STA15D wait after PR-AB	STA16 await TDISind	STA18 await GTA	STA19 await recovery(init)	STA20 await recovery	STA21 await CDA	STA22 await SCDrsp	STA713 data transfer
ED	p48 STA15C	STA15D	STA16		p50 SUERind STA19				p50 SUERind STA713 p51 SUERind STA20
ER	p48 STA15C	STA15D	STA16		p50 SPERind STA19		p48 SPERind STA20		p50 SPERind STA713 p51 SPERind STA20
GT	p59&p185 STA15C p59&¬p185 SGTind [11] STA15C	STA15D	STA16		p60 SGTind [11] STA19 p61 SGTind [11] STA713	p60 SGTind [11] STA20 p61 SGTind [11] STA713	p59 SGTind [11] STA21		p59 SGTind [11] STA713
GTA		STA15D	STA16	STA713 [6]					
GTC		STA15D	STA16						p62 SCGind GTA [11] STA713
PT	p53&p185 STA15C p53&¬p185 SPTind STA15C	STA15D	STA16	p53 SPTind STA18	p53 STA19	p53 STA20	p53 SPTind STA21		p53 SPTind STA713
SCGreq		STA15D							p55 GTC [11] STA18
SGTreq		STA15D				p57 GT [11] STA20 p58 GT [11] STA713			p54 GT [11] STA713
SPTreq		STA15D						p53 PT STA22	p53 PT STA713
SUERreq		STA15D							p50 ED STA19

Table A.15 – Connection release state table without the symmetric synchronize functional unit

State / Event	STA01A await AA	STA01C idle TC con	STA01D await CDO	STA03 await DN	STA05A await PR or RA	STA06 await RS after coll	STA09 await SRELrsp	STA15B wait after PR-RS
DN	STA01A	TDISreq STA01	TDISreq STA01	¬p66 SRELcnf+ TDISreq STA01 p66 SRELcnf+ STA01C			p69&¬p01 SRELcnf+ [32] STA09	
FN-nr	STA01A	TDISreq STA01	TDISreq STA01	¬p65 SRELind [8][18] STA09	p68 STA05A	p68 STA06		
FN-r	STA01A	TDISreq STA01	TDISreq STA01	¬p65&¬p01&p16 SRELind [8][18] STA09	p68&¬p01&p16 STA05A			
NF	STA01A	TDISreq STA01	TDISreq STA01	p67 SRELcnf– STA713				p67 SRELcnf– STA15B
SRELreq							¬p65 FN-nr [8][18] STA09	p63 STA15B
SRELrsp+							¬p66&p75 DN[4] STA16 p66 DN STA01C p69&p01 DN STA03	
SRELrsp–							p67 NF STA713	

Table A.15 *(concluded)* – Connection release state table without the symmetric synchronize functional unit

State / Event	STA15C wait after PR-RA	STA15D wait after PR-AB	STA16 await TDISind	STA19 await recovery(init)	STA20 await recovery	STA713 data transfer
DN			STA16			
FN-nr	p68 STA15C	STA15D	STA16	p68 STA19	p68 STA20	p68 SRELind [8] STA09
FN-r	p68&¬p01&p16 STA15C		STA16	p68&¬p01&p16 STA19	p68&¬p01&p16 STA20	p68&¬p01&p16 SRELind [9] STA09
NF		STA15D	STA16			
SRELreq		STA15D				p63&¬p64 FN-nr [8] STA03 p63&p64 FN-r [7] STA03
SRELrsp+		[4] STA16				
SRELrsp–		STA15D				

Table A.16 – Abort state table without the symmetric synchronize functional unit

State / Event	STA01 idle no TC	STA01A await AA	STA01B await TCONcnf	STA01C idle TC con	STA01D await CDO	STA02A await AC	STA02B await OA	STA03 await DN	STA04A await PR or MAA
AA	//	[3] STA01C	//	TDISreq STA01	TDISreq STA01				
AB-nr	//	[3] TDISreq STA01	//	TDISreq STA01	TDISreq STA01	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA [4] STA16	SxABind(3) TDISreq STA01	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA [4] STA16	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA [4] STA16
AB-r	//	[3] STA01C	//	¬p02 TDISreq STA01 p02 AA STA01C	¬p02 TDISreq STA01 p02 AA STA01C	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C
PR-AB	//	//	//	TDISreq STA01	//	STA15D	//	STA15D	STA15D
SUABreq			TDISreq STA01			¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A
TDISind	//	[3] STA01	SPABind STA01	STA01	STA01	SPABind STA01	SPABind STA01	SPABind STA01	SPABind STA01
TIM	//	TDISreq STA01	//	//	//	//	//	//	//

Table A.16 *(continued)* – Abort state table without the symmetric synchronize functional unit

State / Event	STA04B await PR or AEA	STA05A await PR or RA	STA05B await PR or AIA	STA05C await PR or ADA	STA06 await RS after coll	STA08 await SCONrsp	STA09 await SRELrsp	STA10A await SSYNMrsp
AA								
AB-nr	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA [4] STA16	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA [4] STA16	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA [4] STA16	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA [4] STA16	SxABind(3) TDISreq STA01	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA [4] STA16	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA [4] STA16	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA [4] STA16
AB-r	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C		¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C
PR-AB	STA15D	STA15D	STA15D	STA15D	STA15D	STA15D	STA15D	STA15D
SUABreq	¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	¬p02 PR-AB(4) AB-nr [4] STA16	¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A
TDISind	SPABind STA01	SPABind STA01	SPABind STA01	SPABind STA01	SPABind STA01	SPABind STA01	SPABind STA01	SPABind STA01
TIM	//	//	//	//	//	//	//	//

Table A.16 *(continued)* – Abort state table without the symmetric synchronize functional unit

State / Event	STA10B await SACTErsp	STA11A await SRSYNrsp	STA11B await SACTIrsp	STA11C await SACTDrsp	STA15A wait after PR-MAA	STA15B wait after PR-RS	STA15C wait after PR-RA
AA							
AB-nr	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA [4] STA16	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA [4] STA16	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA [4] STA16	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA [4] STA16	SxABind(3) TDISreq STA01	SxABind(3) TDISreq STA01	SxABind(3) TDISreq STA01
AB-r	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C			
PR-AB	STA15D	STA15D	STA15D	STA15D	STA15D	STA15D	STA15D
SUABreq	¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	¬p02 PR-AB(4) AB-nr [4] STA16	¬p02 PR-AB(4) AB-nr [4] STA16	¬p02 PR-AB(4) AB-nr [4] STA16
TDISind	SPABind STA01	SPABind STA01	SPABind STA01	SPABind STA01	SPABind STA01	SPABind STA01	SPABind STA01
TIM	//	//	//	//	//	//	//

Table A.16 *(concluded)* – Abort state table without the symmetric synchronize functional unit

State / Event	STA15D wait after PR-AB	STA16 await TDISind	STA18 await GTA	STA19 await recovery(init)	STA20 await recovery	STA21 await CDA	STA22 await SCDrsp	STA713 data transfer
AA		[3] TDISreq STA01						
AB-nr	SxABInd(3) TDISreq STA01	[3] TDISreq STA01	¬p02 SxABInd(3) TDISreq STA01 p02 SxABInd(3) AA [4] STA16	¬p02 SxABInd(3) TDISreq STA01 p02 SxABInd(3) AA [4] STA16	¬p02 SxABInd(3) TDISreq STA01 p02 SxABInd(3) AA [4] STA16	¬p02 SxABInd(3) TDISreq STA01 p02 SxABInd(3) AA [4] STA16	¬p02 SxABInd(3) TDISreq STA01 p02 SxABInd(3) AA [4] STA16	¬p02 SxABInd(3) TDISreq STA01 p02 SxABInd(3) AA [4] STA16
AB-r		[3] TDISreq STA01	¬p02 SxABInd(3) TDISreq STA01 p02 SxABInd(3) AA STA01C	¬p02 SxABInd(3) TDISreq STA01 p02 SxABInd(3) AA STA01C	¬p02 SxABInd(3) TDISreq STA01 p02 SxABInd(3) AA STA01C	¬p02 SxABInd(3) TDISreq STA01 p02 SxABInd(3) AA STA01C	¬p02 SxABInd(3) TDISreq STA01 p02 SxABInd(3) AA STA01C	¬p02 SxABInd(3) TDISreq STA01 p02 SxABInd(3) AA STA01C
PR-AB		[3] TDISreq STA01	STA15D	STA15D	STA15D	STA15D	STA15D	STA15D
SUABreq	[4] STA16		¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A
TDISind	SPABind STA01	[3] STA01	SPABind STA01	SPABind STA01	SPABind STA01	SPABind STA01	SPABind STA01	SPABind STA01
TIM	TDISreq STA01	TDISreq STA01	//	//	//	//	//	//

Table A.17 – Connection establishment state table with the symmetric synchronize functional unit

State / Event	STA01 idle no TC	STA01A await AA	STA01B await TCONcnf	STA01C idle TC con	STA01D await CDO	STA02A await AC	STA02B await OA	STA08 await SCONrsp	STA15D wait after PR-AB	STA16 await TDISind
AC	//	STA01A	//	TDISreq STA01		¬p178 SCONcnf+ [5][11] STA713 [6] p178 SCONcnf+ [71][11] STA713 [6]			STA15D	STA16
CDO	//		//	TDISreq STA01	¬p202 [50] STA01D p202 SCONind STA08				STA15D	
CN	//	TDISreq [3] STA01	//	¬p01&¬p76&p204 OA [50] STA01D ¬p01&¬p76&¬p204 SCONind STA08 ¬p01&p76&¬p02 RF-nr [4] STA16 ¬p01&p76&p02 RF-r STA01C p01 TDISreq STA01						TDISreq [3] STA01
OA	//		//	TDISreq STA01			CDO [51] STA02A		STA15D	
RF-nr	//	STA01A	//	TDISreq STA01		SCONcnf– TDISreq STA01	SCONcnf– TDISreq STA01			STA16
RF-r	//	STA01A	//	TDISreq STA01		¬p02 SCONcnf– TDISreq STA01 p02 SCONcnf– STA01C	¬p02 SCONcnf– TDISreq STA01 p02 SCONcnf– STA01C			STA16

Table A.17 (*concluded*) – Connection establishment state table with the symmetric synchronize functional unit

State / Event	STA01 idle no TC	STA01A await AA	STA01B await TCONcnf	STA01C idle TC con	STA01D await CDO	STA02A await AC	STA02B await OA	STA08 await SCONrsp	STA15D wait after PR-AB	STA16 await TDISind
SCONreq	TCONreq [2] STA01B			p01&p204 CN STA02B p01&¬p204 CN STA02A						
SCONrsp+								¬p178 AC [5][11] STA713 p178 AC [71][11] STA713	STA15D	
SCONrsp–								¬p02 RF-nr [4] STA16 p02 RF-r STA01C	[4] STA16	
TCONcnf	//	//	p204 CN STA02B ¬p204 CN STA02A	//	//	//	//	//	//	//
TCONind	TCONrsp [1] STA01C	//	//	//	//	//	//	//	//	//

Table A.18 – Data transfer state table with the symmetric synchronize functional unit

State / Event	STA01A await AA	STA01C idle TC con	STA01D await CDO	STA02A await AC	STA03 await DN	STA04A await PR or MAA	STA04B await PR or AEA	STA05A await PR or RA	STA05B await PR or AIA
DT	STA01A	TDISreq STA01	TDISreq STA01		p05&p10 SDTind STA03	p05 SDTind STA04A	p05 SDTind STA04B	p05&p185 STA05A p05&¬p185 SDTind STA05A	p05 STA05B
EX	STA01A	TDISreq STA01	TDISreq STA01	[10] STA02A	p09 SEXind STA03	p08 SEXind STA04A	p08 SEXind STA04B	p08&p185 STA05A p08&¬p185 SEXind STA05A	p08 STA05B
TD	STA01A	TDISreq STA01	TDISreq STA01		p06&p10 STDind STA03	p06 STDind STA04A	p06 STDind STA04B	p06&p185 STA05A p06&¬p185 STDind STA05A	p06 STA05B
SDTreq								p03&¬p186 DT STA05A	
SEXreq								p08&¬p186 EX STA05A	
STDreq								p06&¬p186 TD STA05A	

Table A.18 *(continued)* – Data transfer state table with the symmetric synchronize functional unit

State / Event	STA05C await PR or ADA	STA06 await RS after coll	STA09 await SRELrsp	STA10A await SSYNMrsp	STA10B await SACTErsp	STA11A await SRSYNrsp	STA15A wait after PR-MAA
DT	p05 STA05C	p05&p185 STA06 p05&¬p185 SDTind STA06				p05&¬p185 SDTind STA11A	p05 SDTind STA15A
EX	p08 STA05C	p08 [10] STA06				p08&¬p185 SEXind STA11A	p08 [10] STA15A
TD	p06 STA05C	p06&p185 STA06 p06&¬p185 STDind STA06				p06&¬p185 STDind STA11A	p06 STDind STA15A
SDTreq		p03&p184&p186 STA06 p03&p184&¬p186 DT STA06	p04 DT STA09	p03 DT STA10A	p03 DT STA10B	p03&¬p186 DT STA11A	
SEXreq		p08&p184&p186 STA06 p08&p184&¬p186 EX STA06	p09 EX STA09	p08 EX STA10A	p08 EX STA10B	p08&¬p186 EX STA11A	
STDreq		p06&p184&p186 STA06 p06&p184&¬p186 TD STA06	p07 TD STA09	p06 TD STA10A	p06 TD STA10B	p06&¬p186 TD STA11A	

Table A.18 *(concluded)* – Data transfer state table with the symmetric synchronize functional unit

State / Event	STA15B wait after PR-RS	STA15C wait after PR-RA	STA15D wait after PR-AB	STA16 await TDISind	STA18 await GTA	STA19 await recovery(init)	STA20 await recovery	STA21 await CDA	STA713 data transfer
DT	p05&p185 STA15B p05&¬p185 SDTind STA15B	p05&p185 STA15C p05&¬p185 SDTind STA15C	STA15D	STA16	p70 SDTind STA18	STA19	p05 STA20	p70 SDTind STA21	p05 SDTind STA713
EX	p08&¬p185 [10] STA15B	p08 [10] STA15C		STA16	p08 SEXind STA18	p08 STA19	p08 STA20	p08 SEXind STA21	p08 SEXind STA713
TD	p06&p185 STA15B p06&¬p185 STDind STA15B	p06&p185 STA15C p06&¬p185 STDind STA15C	STA15D	STA16	p06 STDind STA18	p06 STA19	p06 STA20	p06 STDind STA21	p06 STDind STA713
SDTreq	p03&p186 STA15B p03&¬p186 DT STA15B	p03&¬p186 DT STA15C	STA15D		p70 DT STA18				p03 DT STA713
SEXreq	p08&p186 STA15B p08&¬p186 EX STA15B	p08&¬p186 EX STA15C	STA15D		p08 EX STA18				p08 EX STA713
STDreq	p06&p186 STA15B p06&¬p186 TD STA15B	p06&¬p186 TD STA15C	STA15D		p06 TD STA18				p06 TD STA713

Table A.19 – Synchronization state table with the symmetric synchronize functional unit

State / Event	STA01A await AA	STA01C idle TC con	STA01D await CDO	STA04A await PR or MAA	STA04B await PR or AEA	STA05A await PR or RA	STA05B await PR or AIA	STA05C await PR or ADA	STA06 await RS after coll
MAA/AEA	STA01A	TDISreq STA01	TDISreq STA01	p16&p20 SSYNMcnf [14][22] STA713 p16&p173 SSYNMcnf [14][62][64] STA713	p16&p20 SACTEcnf [14][22] STA713 p16&p173 SACTEcnf [14][62][64] STA713	[43] STA05A	[43] STA05B	[43] STA05C	[43] STA06
MAP	STA01A	TDISreq STA01	TDISreq STA01			p12&p185 STA05A p12&p178&¬p185 SSYNMind [62] STA05A p12&¬p178&¬p185 SSYNMind [23] STA05A			p12&p185 STA06 p12&p178&¬p185 SSYNMind [62] STA06 p12&¬p178&¬p185 SSYNMind [23] STA06
PR-MAA	STA01A	TDISreq STA01	TDISreq STA01	STA15A	STA15A	STA05A	STA05B	STA05C	
SSYNMreq									
SSYNMrsp									

Table A.19 *(continued)* – Synchronization state table with the symmetric synchronize functional unit

State / Event	STA10A await SSYNMrsp	STA15A wait after PR-MAA	STA15B wait after PR-RS	STA15C wait after PR-RA	STA15D wait after PR-AB	STA16 await TDISind	STA19 await recovery(init)	STA20 await recovery	STA713 data transfer
MAA/AEA		p20&¬p23 SSYNMcnf [14][22] STA713 [6] p20&p23 SACTEcnf [14][22] STA713 [6] p174&¬p23 SSYNMcnf [14][62][64] STA713 [6] p174&p23 SACTEcnf [14][62][64] STA713 [6]	[43] STA15B	[43] STA15C	STA15D	STA16		p20 STA20	
MAP			p12 STA15B	p12&p185 STA15C p12&p178&¬p185 SSYNMind [62] STA15C p12&¬p178&¬p185 SSYNMind [23] STA15C	STA15D	STA16	p12&p19 [31] STA19 p12&p177 [62] STA19	p12&p19 [31] STA20 p12&p177 [62] STA20	p12&p19 SSYNMind [13][31] STA10A p12&p177 SSYNMind [13][62] STA10A
PR-MAA						STA16			
SSYNMreq			p13&p186 STA15B p13&¬p186 MAP [63] STA15B		STA15D				p13&¬p178 MAP [13][24] STA04A p13&p178 MAP [13][63] STA04A
SSYNMrsp	¬p178 PR-MAA(1) MAA [14][22] STA713 p178 PR-MAA(1) MAA [14][63][64] STA713		STA15B		STA15D				

Table A.19 *(continued)* – Synchronization state table with the symmetric synchronize functional unit

State / Event	STA01A await AA	STA01C idle TC con	STA01D await CDO	STA03 await DN	STA04A await PR or MAA	STA04B await PR or AEA	STA05A await PR or RA
AE	STA01A	TDISreq STA01	TDISreq STA01				p72 STA05A
MIA	STA01A	TDISreq STA01	TDISreq STA01	p17&p21 SSYNmcnf [25] STA03 p17&p176 SSYNmcnf [65] STA03	p17&¬p20&p21 SSYNmcnf [25] STA04A p17&p176 SSYNmcnf [65] STA04A	p17&¬p20&p21 SSYNmcnf [25] STA04B p17&p176 SSYNmcnf [65] STA04B	p17&¬p81&p186 STA05A p17&p21&p81 SSYNmcnf [25] STA05A p17&p176&p81 SSYNmcnf [65] STA05A p17&¬p81&p176&p186 SSYNmcnf [65] STA05A
MIP	STA01A	TDISreq STA01	TDISreq STA01	p178&p26 SSYNmind [62] STA03	p178&p26 SSYNmind [62] STA04A	p178&p26 SSYNmind [62] STA04B	p14&¬p178&p185 STA05A p178&p185&p26 STA05A p178&¬p185&p26 SSYNmind [62] STA05A ¬p178&¬p185&p26 SSYNmind [23] STA05A
MIP-d	STA01A	TDISreq STA01	TDISreq STA01	p82&p178&p26 SSYNmind [44][62] STA03	p82&p178&p26 SSYNmind [44][62] STA04A	p82&p178&p26 SSYNmind [44][62] STA04B	p14&p82 [44] STA05A p82&p178&p185&p26 [44] STA05A p82&p178&¬p185&p26 SSYNmind [44][62] STA05A p82&¬p178&¬p185&p26 SSYNmind [44][23] STA05A

Table A.19 *(continued)* – Synchronization state table with the symmetric synchronize functional unit

State / Event	STA05B await PR or AIA	STA05C await PR or ADA	STA06 await RS after coll
AE			p72 STA06
MIA	p17 STA05B	p17 STA05C	p17&¬p81&p186 STA06 p17&p21&p81 SSYNmcnf [25] STA06 p17&p176&p81 SSYNmcnf [65] STA06 p17&¬p81&p176&p186 SSYNmcnf [65] STA06
MIP	p14 STA05B p178 STA05B	p14 STA05C p178 STA05C	p14 STA06 p178&p185&p26 STA06 p178&¬p185&p26 SSYNmind [62] STA06 ¬p178&¬p185&p26 SSYNmind [23] STA06
MIP-d	p14&p82 [44] STA05B p82&p178 [44] STA05B	p14&p82 [44] STA05C p82&p178 [44] STA05C	p14&p82 [44] STA06 p82&p178&p185&p26 [44] STA06 p82&p178&¬p185&p26 SSYNmind [44][62] STA06 p82&¬p178&¬p185&p26 SSYNmind [44][23] STA06

Table A.19 *(continued)* – Synchronization state table with the symmetric synchronize functional unit

State / Event	STA11A await SRSYNrsp	STA15A wait after PR-MAA	STA15B wait after PR-RS	STA15C wait after PR-RA
AE			p72 STA15B	p72 STA15C
MIA	¬p186 SSYNmcnf [65] STA11A	p17&¬p20&p21 SSYNmcnf [25] STA15A p17&p176 SSYNmcnf [65] STA15A	p17&¬p81&p186 STA15B p17&p21&p81 SSYNmcnf [25] STA15B p17&p176&p81 SSYNmcnf [65] STA15B p17&¬p81&p176&p186 SSYNmcnf [65] STA15B	p17&¬p81&p186 STA15C p17&p21&p81 SSYNmcnf [25] STA15C p17&p176&p81 SSYNmcnf [65] STA15C p17&¬p81&p176&p186 SSYNmcnf [65] STA15C p17&¬p186 SSYNmcnf [65] STA15C
MIP	¬p185 SSYNmind [62] STA11A	p178 SSYNmind [62] STA15A	p14&¬p178&p185 STA15B p178&p185 STA15B p178&¬p185 SSYNmind [62] STA15B p14&¬p178&¬p185 SSYNmind [23] STA15B	p14&¬p178&p185 STA15C p178&p185 STA15C p178&¬p185 SSYNmind [62] STA15C p14&¬p178&¬p185 SSYNmind [23] STA15C
MIP-d		p82&p178 SSYNmind [44][62] STA15A		p14&p82&¬p178&p185 [44] STA15C p14&p82&¬p178&¬p185 SSYNmind [44][23] STA15C p82&p178&p185 [44] STA15C p82&p178&¬p185 SSYNmind [44][62] STA15C

Table A.19 *(concluded)* – Synchronization state table with the symmetric synchronize functional unit

State / Event	STA15D wait after PR-AB	STA16 await TDISind	STA19 await recovery(init)	STA20 await recovery	STA713 data transfer
AE	STA15D	STA16	p72&p19 [31] STA19 p72&p177 [62] STA19	p72&p19 [31] STA20 p72&p177 [62] STA20	p72&p19 SACTEind [13][31] STA10B p72&p177 SACTEind [13][62] STA10B
MIA	STA15D	STA16	p17&p21 [25] STA19 p17&p176 [65] STA19	p17&p21 STA20 p17&p176 [65] STA20	p17&p21 SSYNmcnf [25] STA713 p17&p176 SSYNmcnf [65] STA713
MIP	STA15D	STA16	p14&p19 [23] STA19 p26&p178 [62] STA19	p14&p19 [23] STA20 p26&p178 [62] STA20	p14&p19 SSYNmind [23] STA713 p26&p178 SSYNmind [62] STA713
MIP-d	p82 [44] STA15D	p82 [44] STA16	p14&p19&p82 [44][23] STA19 p26&p82&p178 [44][62] STA19	p14&p19&p82 [44][23] STA20 p26&p82&p178 [44][62] STA20	p14&p19&p82 SSYNmind [44][23] STA713 p26&p82&p178 SSYNmind [44][62] STA713

Table A.19 *(continued)* – Synchronization state table with the symmetric synchronize functional unit

State / Event	STA05A await PR or RA
SACTEreq	
SACTErsp	
SSYNmdreq	p82&¬p186 MIP [42][63] STA05A
SSYNmreq	¬p186 MIP [63] STA05A
SSYNmrsp	p179&¬p185 MIA [66] STA05A

Table A.19 *(continued)* – Synchronization state table with the symmetric synchronize functional unit

State / Event	STA06 await RS after coll	STA09 await SRELrsp	STA10A await SSYNMrsp
SACTEreq			
SACTErsp			
SSYNmdreq	p82&p184&p186 STA06 p82&p184&¬p186 MIP [42][63] STA06	p82&p178&p26 MIP [42][63] STA09	p82&p178&p26 MIP [42][63] STA10A
SSYNmreq	p184&p186 STA06 p184&¬p186 MIP [63] STA06	p178&p26 MIP [63] STA09	p178&p26 MIP [63] STA10A
SSYNmrsp	p179&p184&p185 STA06 p179&p184&¬p185 MIA [66] STA06	p18&p21 MIA [25] STA09 p18&p179 MIA [66] STA09	p18&¬p20&p21 MIA [25] STA10A p18&¬p175&p179 MIA [66] STA10A

Table A.19 *(continued)* – Synchronization state table with the symmetric synchronize functional unit

State / Event	STA10B await SACTErsp	STA11A await SRSYNrsp	STA15B wait after PR-RS	STA15C wait after PR-RA
SACTEreq			p71 STA15B	
SACTErsp	¬p178 PR-MAA(1) AEA [14][22] STA713 p178 PR-MAA(1) AEA [14][62][64] STA713			
SSYNmdreq	p82&p178&p26 MIP [42][63] STA10B	p82&¬p186 MIP [42][63] STA11A	p15&p82 STA15B p82&p178&p186 STA15B p82&¬p186 MIP [42][63] STA15B	p82&¬p186 MIP [42][63] STA15C
SSYNmreq	p178&p26 MIP [63] STA10B	¬p186 MIP [63] STA11A	p15 STA15B p178&p186 STA15B ¬p186 MIP [63] STA15B	¬p186 MIP [63] STA15C
SSYNmrsp	p18&¬p20&p21 MIA [25] STA10B p18&¬p175&p179 MIA [66] STA10B	p179&¬p185 MIA [66] STA11A	p18&p21 STA15B p179&p185 STA15B p179&¬p185 MIA [66] STA15B	p179&¬p185 MIA [66] STA15C

Table A.19 (*concluded*) – Synchronization state table with the symmetric synchronize functional unit

State / Event	STA15D wait after PR-AB	STA713 data transfer
SACTEreq	STA15D	p71&¬p178 AE [13][24] STA04B p71&p178 AE [13][63] STA04B
SACTErsp	STA15D	
SSYNmdreq	STA15D	p15&p82 MIP [41][24] STA713 p82&p178&p26 MIP [42][63] STA713
SSYNmreq	STA15D	p15 MIP [24] STA713 p178&p26 MIP [63] STA713
SSYNmrsp	STA15D	p18&p21 MIA [25] STA713 p26&p179 MIA [66] STA713

Table A.20 – Resynchronization state table with the symmetric synchronize functional unit

State / Event	STA01A await AA	STA01C idle TC con	STA01D await CDO	STA02A await AC	STA03 await DN	STA04A await PR or MAA	STA04B await PR or AEA	STA05A await PR or RA
PR-RA	STA01A	TDISreq STA01	TDISreq STA01					STA15C
PR-RS	STA01A	TDISreq STA01	TDISreq STA01	[10] STA02A	p10&p80 [67] STA15B	p80 [67] STA15B	p80 [67] STA15B	p80 [67] STA06
RA	STA01A	TDISreq STA01	TDISreq STA01					p35&p49 SRSYNcnf [28][11] STA713 p35&p187 SRSYNcnf [68][11] STA713
RS	STA01A	TDISreq STA01			p10&¬p34&p35&p32 [19] SRSYNind(2) [16] STA11A	p35&p32 [19] SRSYNind(2) [16] STA11A	p35&p32 [19] SRSYNind(2) [16] STA11A	¬p24&p35&p32 STA05A [72] p24&p35&p32 [19] SRSYNind(2) [16] STA11A
SRSYNreq						p28&p180 PR-RS(5) RS [16] STA05A		
SRSYNrsp								

Table A.20 *(continued)* – Resynchronization state table with the symmetric synchronize functional unit

State / Event	STA05B await PR or AIA	STA05C await PR or ADA	STA06 await RS after coll	STA09 await SRELrsp	STA10A await SSYNMrsp	STA10B await SACTErsp	STA11A await SRSYNrsp
PR-RA	STA15C	STA15C	[10] STA06				
PR-RS	STA05B	STA05C	[10] STA06		p80 [67] STA15B	p80 [67] STA15B	
RA							
RS	p35 STA05B	p35 STA05C	¬p24&p32 STA05A [72][6] p24&p32 [19] SRSYNind(2) [16] STA11A [6]		p35&p180 [19] SRSYNind(2) [16] STA11A		
SRSYNreq				p10&p28&¬p34&p32 PR-RS(5) RS [16] STA05A	p28&p32 PR-RS(5) RS [16] STA05A	p28&p32 PR-RS(5) RS [16] STA05A	p24&p32 PR-RS(5) RS [16] STA05A
SRSYNrsp							p43 PR-RA(1) RA [28][11] STA713 p187 PR-RA(1) RA [68][11] STA713

Table A.20 *(concluded)* – Resynchronization state table with the symmetric synchronize functional unit

State / Event	STA15A wait after PR-MAA	STA15B wait after PR-RS	STA15C wait after PR-RA	STA15D wait after PR-AB	STA16 await TDISind	STA18 await GTA	STA19 await recovery(init)	STA20 await recovery	STA713 data transfer
PR-RA					STA16				
PR-RS	[10] STA15A		[10] STA15C		STA16	[10] STA18	p80 [67] STA15B	p80 [67] STA15B	p26 [67] STA15B ¬p26 [10] STA713
RA			p36&p49 SRSYNcnf [28][11] STA713 [6] p36&p187 SRSYNcnf [68][11] STA713 [6]	STA15D	STA16				
RS		p29&p32 [19] SRSYNind(2) [16] STA11A			STA16		p35&p32 [19] SRSYNind(2) [16] STA11A	p35&p32 [19] SRSYNind(2) [16] STA11A	p26&p35&p32 [19] SRSYNind(2) [16] STA11A
SRSYNreq	p28&p30&p180 PR-RS(5) RS [16] STA05A	p27&p28&p32 PR-RS(5) RS [16] STA06						p28&p32 PR-RS(5) RS [16] STA05A	p29&p32 PR-RS(5) RS [16] STA05A
SRSYNrsp				STA15D					

Table A.21 – Activity interrupt and discard state table with the symmetric synchronize functional unit

State / Event	STA01A await AA	STA01C idle TC con	STA01D await CDO	STA04A await PR or MAA	STA04B await PR or AEA	STA05A await PR or RA	STA05B await PR or AIA	STA05C await PR or ADA	STA06 await RS after coll
AD	STA01A	TDISreq STA01	TDISreq STA01			p38&p40 SACTDind [16] STA11C			p37&p40 SACTDind [16] STA11C
ADA	STA01A	TDISreq STA01	TDISreq STA01					p38 SACTDcnf [29] STA713	
AI	STA01A	TDISreq STA01	TDISreq STA01			p38&p40 SACTIind [16] STA11B			p37&p40 SACTIind [16] STA11B
AIA	STA01A	TDISreq STA01	TDISreq STA01				p38 SACTIcnf [29] STA713		
SACTDreq				p34&p39 PR-RS(1) AD [16] STA05C	p39 PR-RS(1) AD [16] STA05C				
SACTDrsp									
SACTIreq				p34&p39 PR-RS(1) AI [16] STA05B	p39 PR-RS(1) AI [16] STA05B				
SACTIrsp									

Table A.21 *(continued)* – Activity interrupt and discard state table with the symmetric synchronize functional unit

State / Event	STA10A await SSYNMrsp	STA10B await SACTErsp	STA11A await SRSYNrsp	STA11B await SACTIrsp	STA11C await SACTDrsp	STA15A wait after PR-MAA	STA15B wait after PR-RS
AD	p38&p40 SACTDind [16] STA11C	p38&p40 SACTDind [16] STA11C					p37&p40 SACTDind [16] STA11C
ADA							
AI	p38&p40 SACTIind [16] STA11B	p38&p40 SACTIind [16] STA11B					p37&p40 SACTIind [16] STA11B
AIA							
SACTDreq			p34&p39 PR-RS(1) AD [16] STA05C			p34&p39 PR-RS(1) AD [16] STA05C [6]	p27&p34&p39 PR-RS(1) AD [16] STA05C
SACTDrsp					PR-RA(1) ADA [30] STA713		
SACTIreq			p34&p39 PR-RS(1) AI [16] STA05B			p34&p39 PR-RS(1) AI [16] STA05B [6]	p27&p34&p39 PR-RS(1) AI [16] STA05B
SACTIrsp				PR-RA(1) AIA [30] STA713			

Table A.21 *(concluded)* – Activity interrupt and discard state table with the symmetric synchronize functional unit

State / Event	STA15C wait after PR-RA	STA15D wait after PR-AB	STA16 await TDISind	STA19 await recovery(init)	STA20 await recovery	STA21 await CDA	STA22 await SCDrsp	STA713 data transfer
AD		STA15D	STA16	p38&p40 SACTDind [16] STA11C	p38&p40 SACTDind [16] STA11C			p38&p40 SACTDind [16] STA11C
ADA	p37&p41 SACTDcnf [29] STA713 [6]	STA15D	STA16					
AI		STA15D	STA16	p38&p40 SACTIind [16] STA11B	p38&p40 SACTIind [16] STA11B		p16 SACTIind [16] STA11B	p38&p40 SACTIind [16] STA11B
AIA	p37&p42 SACTIcnf [29] STA713 [6]	STA15D	STA16					
SACTDreq		STA15D			p34&p11 PR-RS(1) AD [16] STA05C			p34&p39 PR-RS(1) AD [16] STA05C
SACTDrsp		STA15D						
SACTIreq		STA15D			p34&p11 PR-RS(1) AI [16] STA05B	p16 AI [16] STA05B		p34&p39 PR-RS(1) AI [16] STA05B
SACTIrsp		STA15D						

Table A.22 – Activity start, resume and capability data state table with the symmetric synchronize functional unit

State / Event	STA01A await AA	STA01C idle TC con	STA01D await CDO	STA05B await PR or AIA	STA15B wait after PR-RS	STA15D wait after PR-AB	STA16 await TDISind	STA21 await CDA	STA22 await SCDrsp	STA713 data transfer
AR	STA01A	TDISreq STA01	TDISreq STA01		p44&¬p178 SACTRind [12][27] STA15B p44&p178 SACTRind [12][69] STA15B	STA15D	STA16			p44&¬p178 SACTRind [12][27] STA713 [6] p44&p178 SACTRind [12][69] STA713 [6]
AS	STA01A	TDISreq STA01	TDISreq STA01		p44&¬p178 SACTSind [12][26] STA15B p44&p178 SACTSind [12][70] STA15B	STA15D	STA16			p44&¬p178 SACTSind [12][26] STA713 [6] p44&p178 SACTSind [12][70] STA713 [6]
CD	STA01A	TDISreq STA01	TDISreq STA01			STA15D	STA16			p46 SCDind STA22
CDA	STA01A	TDISreq STA01	TDISreq STA01	¬p26 STA05B		STA15D	STA16	SCDcnf STA713		
SACTRreq						STA15D				p45&¬p178 AR [12][27] STA713 p45&p178 AR [12][69] STA713
SACTSreq										p45&¬p178 AS [12][26] STA713 p45&p178 AS [12][70] STA713
SCDreq						STA15D				p47 CD STA21
SCDrsp						STA15D			CDA STA713	

Table A.23 – Token management and exceptions state table with the symmetric synchronize functional unit

State / Event	STA01A await AA	STA01C idle TC con	STA01D await CDO	STA03 await DN	STA04A await PR or MAA	STA04B await PR or AEA	STA05A await PR or RA	STA05B await PR or AIA
ED	STA01A	TDISreq STA01	TDISreq STA01	p52 SUERind STA20	p48&p03 SUERind STA20 p48&¬p03 SUERind STA713	p48&p03 SUERind STA20 p48&¬p03 SUERind STA713	p48 STA05A	p48 STA05B
ER	STA01A	TDISreq STA01	TDISreq STA01	p52 SPERind STA20	p48&p03 SPERind STA20 p48&¬p03 SPERind STA713	p48&p03 SPERind STA20 p48&¬p03 SPERind STA713	p48 STA05A	p48 STA05B
GT	STA01A	TDISreq STA01	TDISreq STA01		p59 SGTind [11] STA04A	p59 SGTind [11] STA04B	p59&p185 STA05A p59&¬p185 SGTind [11] STA05A	p59 STA05B
GTA	STA01A	TDISreq STA01	TDISreq STA01					
GTC	STA01A	TDISreq STA01	TDISreq STA01					
PT	STA01A	TDISreq STA01	TDISreq STA01	p53 SPTind STA03	p53 SPTind STA04A	p53 SPTind STA04B	p53&p185 STA05A p53&¬p185 SPTind STA05A	p53 STA05B
SCGreq								
SGTreq					p54 GT [11] STA04A	p54 GT [11] STA04B		
SPTreq								
SUERreq								

Table A.23 *(continued)* – Token management and exceptions state table with the symmetric synchronize functional unit

State / Event	STA05C await PR or ADA	STA06 await RS after coll	STA09 await SRELrsp	STA10A await SSYNMrsp	STA10B await SACTErsp	STA15A wait after PR-MAA	STA15B wait after PR-RS
ED	p48 STA05C	p48 STA06					
ER	p48 STA05C	p48 STA06					
GT	p59 STA05C	p59&p185 STA06 p59&¬p185 SGTind [11] STA06		p59 SGTind [11] STA10A	p59 SGTind [11] STA10B	p59 SGTind [11] STA15A	p59&p185 STA15B p59&¬p185 SGTind [11] STA15B
GTA							
GTC							
PT	p53 STA05C	p53&p185 STA06 p53&¬p185 SPTind STA06				p53 SPTind STA15A	p53&p185 STA15B p53&¬p185 SPTind STA15B
SCGreq							
SGTreq				p54 GT [11] STA10A	p54 GT [11] STA10B	p54 GT [11] STA15A	p54 STA15B
SPTreq			p53 PT STA09	p53 PT STA10A	p53 PT STA10B		p53 STA15B
SUERreq			p50 ED STA19	p50 ED STA19	p50 ED STA19		p50 STA15B

Table A.23 *(concluded)* – Token management and exceptions state table with the symmetric synchronize functional unit

State / Event	STA15C wait after PR-RA	STA15D wait after PR-AB	STA16 await TDISind	STA18 await GTA	STA19 await recovery(init)	STA20 await recovery	STA21 await CDA	STA22 await SCDrsp	STA713 data transfer
ED	p48 STA15C	STA15D	STA16		p50 SUERind STA19				p50 SUERind STA713 p51 SUERind STA20
ER	p48 STA15C	STA15D	STA16		p50 SPERind STA19		p48 SPERind STA20		p50 SPERind STA713 p51 SPERind STA20
GT	p59&p185 STA15C p59&¬p185 SGTind [11] STA15C	STA15D	STA16		p60 SGTind [11] STA19 p61 SGTind [11] STA713	p60 SGTind [11] STA20 p61 SGTind [11] STA713	p59 SGTind [11] STA21		p59 SGTind [11] STA713
GTA		STA15D	STA16	STA713 [6]					
GTC		STA15D	STA16						p62 SCGind GTA [11] STA713
PT	p53&p185 STA15C p53&¬p185 SPTind STA15C	STA15D	STA16	p53 SPTind STA18	p53 STA19	p53 STA20	p53 SPTind STA21		p53 SPTind STA713
SCGreq		STA15D							p55 GTC [11] STA18
SGTreq		STA15D				p57 GT [11] STA20 p58 GT [11] STA713			p54 GT [11] STA713
SPTreq		STA15D						p53 PT STA22	p53 PT STA713
SUERreq		STA15D							p50 ED STA19

Table A.24 – Connection release state table with the symmetric synchronize functional unit

State / Event	STA01A await AA	STA01C idle TC con	STA01D await CDO	STA03 await DN	STA05A await PR or RA	STA06 await RS after coll	STA09 await SRELrsp	STA15B wait after PR-RS
DN	STA01A	TDISreq STA01	TDISreq STA01	¬p66 SRELcnf+ TDISreq STA01 p66 SRELcnf+ STA01C			p69&¬p01 SRELcnf+ [32] STA09	
FN-nr	STA01A	TDISreq STA01	TDISreq STA01	¬p65 SRELind [8][18] STA09	p68 STA05A	p68 STA06		
FN-r	STA01A	TDISreq STA01	TDISreq STA01	¬p65&¬p01&p16 SRELind [8][18] STA09	p68&¬p01&p16 STA05A			
NF	STA01A	TDISreq STA01	TDISreq STA01	p67 SRELcnf– STA713				p67 SRELcnf– STA15B
SRELreq							¬p65 FN-nr [8][18] STA09	p63 STA15B
SRELrsp+							¬p66&p75 DN[4] STA16 p66 DN STA01C p69&p01 DN STA03	
SRELrsp–							p67 NF STA713	

Table A.24 *(concluded)* – Connection release state table with the symmetric synchronize functional unit

State / Event	STA15C wait after PR-RA	STA15D wait after PR-AB	STA16 await TDISind	STA19 await recovery(init)	STA20 await recovery	STA713 data transfer
DN			STA16			
FN-nr	p68 STA15C	STA15D	STA16	p68 STA19	p68 STA20	p68 SRELind [8] STA09
FN-r	p68&¬p01&p16 STA15C		STA16	p68&¬p01&p16 STA19	p68&¬p01&p16 STA20	p68&¬p01&p16 SRELind [9] STA09
NF		STA15D	STA16			
SRELreq		STA15D				p63&¬p64 FN-nr [8] STA03 p63&p64 FN-r [7] STA03
SRELrsp+		[4] STA16				
SRELrsp–		STA15D				

Table A.25 – Abort state table with the symmetric synchronize functional unit

State / Event	STA01 idle no TC	STA01A await AA	STA01B await TCONcnf	STA01C idle TC con	STA01D await CDO	STA02A await AC	STA02B await OA	STA03 await DN	STA04A await PR or MAA
AA	//	[3] STA01C	//	TDISreq STA01	TDISreq STA01				
AB-nr	//	[3] TDISreq STA01	//	TDISreq STA01	TDISreq STA01	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA [4] STA16	SxABind(3) TDISreq STA01	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA [4] STA16	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA [4] STA16
AB-r	//	[3] STA01C	//	¬p02 TDISreq STA01 p02 AA STA01C	¬p02 TDISreq STA01 p02 AA STA01C	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C
PR-AB	//	//	//	TDISreq STA01	//	STA15D	//	STA15D	STA15D
SUABreq			TDISreq STA01			¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A
TDISind	//	[3] STA01	SPABind STA01	STA01	STA01	SPABind STA01	SPABind STA01	SPABind STA01	SPABind STA01
TIM	//	TDISreq STA01	//	//	//	//	//	//	//

Table A.25 *(continued)* – Abort state table with the symmetric synchronize functional unit

State / Event	STA04B await PR or AEA	STA05A await PR or RA	STA05B await PR or AIA	STA05C await PR or ADA	STA06 await RS after coll	STA08 await SCONrsp	STA09 await SRELrsp	STA10A await SSYNMrsp
AA								
AB-nr	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA [4] STA16	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA [4] STA16	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA [4] STA16	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA [4] STA16	SxABind(3) TDISreq STA01	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA [4] STA16	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA [4] STA16	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA [4] STA16
AB-r	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C		¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C
PR-AB	STA15D	STA15D	STA15D	STA15D	STA15D	STA15D	STA15D	STA15D
SUABreq	¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	¬p02 PR-AB(4) AB-nr [4] STA16	¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A
TDISind	SPABind STA01	SPABind STA01	SPABind STA01	SPABind STA01	SPABind STA01	SPABind STA01	SPABind STA01	SPABind STA01
TIM	//	//	//	//	//	//	//	//

Table A.25 *(continued)* – Abort state table with the symmetric synchronize functional unit

State / Event	STA10B await SACTErsp	STA11A await SRSYNrsp	STA11B await SACTIrsp	STA11C await SACTDrsp	STA15A wait after PR-MAA	STA15B wait after PR-RS	STA15C wait after PR-RA
AA							
AB-nr	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA [4] STA16	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA [4] STA16	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA [4] STA16	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA [4] STA16	SxABind(3) TDISreq STA01	SxABind(3) TDISreq STA01	SxABind(3) TDISreq STA01
AB-r	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C			
PR-AB	STA15D	STA15D	STA15D	STA15D	STA15D	STA15D	STA15D
SUABreq	¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	¬p02 PR-AB(4) AB-nr [4] STA16	¬p02 PR-AB(4) AB-nr [4] STA16	¬p02 PR-AB(4) AB-nr [4] STA16
TDISind	SPABind STA01	SPABind STA01	SPABind STA01	SPABind STA01	SPABind STA01	SPABind STA01	SPABind STA01
TIM	//	//	//	//	//	//	//

Table A.25 *(concluded)* – Abort state table with the symmetric synchronize functional unit

State / Event	STA15D wait after PR-AB	STA16 await TDISind	STA18 await GTA	STA19 await recovery(init)	STA20 await recovery	STA21 await CDA	STA22 await SCDrsp	STA713 data transfer
AA		[3] TDISreq STA01						
AB-nr	SxABind(3) TDISreq STA01	[3] TDISreq STA01	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA [4] STA16	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA [4] STA16	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA [4] STA16	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA [4] STA16	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA [4] STA16	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA [4] STA16
AB-r		[3] TDISreq STA01	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C	¬p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C
PR-AB		[3] TDISreq STA01	STA15D	STA15D	STA15D	STA15D	STA15D	STA15D
SUABreq	[4] STA16		¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	¬p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A
TDISind	SPABind STA01	[3] STA01	SPABind STA01	SPABind STA01	SPABind STA01	SPABind STA01	SPABind STA01	SPABind STA01
TIM	TDISreq STA01	TDISreq STA01	//	//	//	//	//	//

NOTES to Tables A.8 to A.25:

1 PR is not sent if TEXP is false.

2 The serial number given in the indication is V(M).

3 SxABind means generate event SUABind if bit 2 of the Transport Disconnect PV field in the ABORT SPDU has the value "user abort". Otherwise, SxABind means generate the event SPABind.

4 PR-AB is only sent if TEXP is true and the SS-user data exceeds 9 octets (see 7.9.2).

5 PR is not sent if p81 is true, or TEXP is false, or, as a local choice, if the data separation functional unit is selected.

Annex B

PGIs and PIs reserved for use by Recommendation T.62

(This annex forms an integral part of this Recommendation | International Standard)

Table B.1 lists the PGIs and PIs which are not defined in this Recommendation | International Standard, but which are reserved as they are used in Recommendation T.62 for parameters that are relevant to higher layers than the session layer.

Table B.1 – PGI and PI reserved for use by Recommendation T.62

PGI		
	0	Reserved for extension
	2	Non-basic session capabilities
PI		
	8	Service identifier
	13	Miscellaneous session capabilities
	14	Window size
	18	Inactivity timer
	28	Initiator's reference number
	29	Acceptor's reference number
	30	Reactivation/transaction indication
	31	Suspend reject reason
PGI		
	32	Reserved for extension
PI		
	40	Service interworking identifier
	44	Acceptance of CDCL parameters
	45	Storage capacity negotiation
	48	Document type identifier
PGI		
	64	Reserved for extension
	65	Non-basic teletex terminal capabilities
PI		
	72	Graphics character set
	73	Control character set
	74	Teletex page format
	75	Miscellaneous teletex terminal capabilities
	77	Number of dots for character box height
	78	Number of dots for character box width

Annex C

Relationship to Recommendation T.62 encoding

(This annex forms an integral part of this Recommendation | International Standard)

This Recommendation | International Standard has been designed to be compatible with Recommendation T.62.

Table C.1 shows the relationship between the T.62 commands and responses and the SPDUs used in this Recommendation | International Standard.

Table C.2 shows the relationship between the T.62 PGI and PI parameters and the PGI and PI parameters used in this Recommendation | International Standard.

Annex B lists the PGIs and PIs which are not defined in this Recommendation | International Standard, but are reserved as they are used in Recommendation T.62 for parameters relevant to higher layers than the session layer. Use of these PGIs and PIs is necessary for correct operation to Recommendation T.62. An implementation of the protocol specified in this Recommendation | International Standard will need to be enhanced to take account of these PGI and PI units.

Table C.1 – Relationship between T.62 commands and responses and ITU-T Rec. X.225 | ISO/IEC 8327-1 SPDUs

Code	T.62 name	SPDU code	SPDU name
13	CSS	CN	CONNECT
15	xxxx	CDO	CONNECT DATA OVERFLOW
16	xxxx	OA	OVERFLOW ACCEPT
14	RSSP	AC	ACCEPT
12	RSSN	RF	REFUSE
9	CSE	FN	FINISH
10	RSEP	DN	DISCONNECT
25	CSA	AB	ABORT
26	RSAP	AA	ABORT ACCEPT
1	CSUI-CDUI	DT	DATA TRANSFER
2	RSUI	PT	PLEASE TOKENS
21	CSCC	GTC	GIVE TOKENS CONFIRM
22	RSCCP	GTA	GIVE TOKENS ACK
1	CSUI	GT	GIVE TOKENS
0	RSUI-RDGR	ER	EXCEPTION REPORT
48	RSUI-RDPBN	ED	EXCEPTION DATA
33	CSTD	TD	TYPED DATA
8	xxxx	NF	NOT FINISHED
49	CSUI-CDPB	MIP	MINOR SYNC POINT
50	RSUI-RDPBP	MIA	MINOR SYNC ACK
41	CSUI-CDE	MAP	MAJOR SYNC POINT
42	RSUI-RDEP	MAA	MAJOR SYNC ACK
7	xxxx	PR	PREPARE
53	xxxx	RS	RESYNCHRONIZE
34	xxxx	RA	RESYNCHRONIZE ACK
5	xxxx	EX	EXPEDITED DATA
45	CSUI-CDS	AS	ACTIVITY START
29	CSUI-CDC	AR	ACTIVITY RESUME
25	CSUI-CDR	AI	ACTIVITY INTERRUPT
26	RSUI-RDRP	AIA	ACTIVITY INTERRUPT ACK
57	CSUI-CDD	AD	ACTIVITY DISCARD
58	RSUI-RDDP	ADA	ACTIVITY DISCARD ACK
41	CSUI-CDE	AE	ACTIVITY END
42	RSUI-RDEP	AEA	ACTIVITY END ACK
61	CSUI-CDCL	CD	CAPABILITY DATA
62	RSUI-RDCLP	CDA	CAPABILITY DATA ACK

Table C.2 – Relationship between T.62 PGI/PI and ITU-T Rec. X.225 | ISO/IEC 8327-1 parameters

T.62 parameters	Code	ITU-T Rec. X.225 | ISO/IEC 8327-1 parameters
PGI		
Reserved for extension	0	(Table B.1)
Session reference	1	Connection Identifier
Non-basic session capabilities	2	(Table B.1)
	3	
	4	
	5	Connect/Accept Item
	6	
	7	
PI		
Service identifier	8	(Table B.1)
Terminal identifier (called terminal)	9	Called SS-user Reference
Terminal identifier (calling terminal)	10	Calling SS-user Reference
Date and time	11	Common Reference
Additional session reference number	12	Additional Reference Information
Miscellaneous session capabilities	13	(Table B.1)
Window size	14	(Table B.1)
	15	Sync Type Item
Session control functions	16	Token Item
Session termination parameter	17	Transport Disconnect
Inactivity timer	18	(Table B.1)
	19	Protocol Options
Session service functions	20	Session Requirement
	21	TSDU Maximum Size
	22	Version Number
	23	Initial Serial Number
	24	Prepare Type
	25	Enclosure Item
	26	Token Setting Item
	27	Resync Type
Initiator's reference number	28	(Table B.1)
Acceptor's reference number	29	(Table B.1)
Reactivation/transaction indication	30	(Table B.1)
Suspend reject reason	31	(Table B.1)
PGI		
Reserved for extension	32	(Table B.1)
Document linking	33	Linking Information
	34	
	35	
	36	
	37	
	38	
	39	

Table C.2 *(concluded)* – Relationship between T.62 PGI/PI and ITU-T Rec. X.225 | ISO/IEC 8327-1 parameters

T.62 parameters	Code	ITU-T Rec. X.225 | ISO/IEC 8327-1 parameters
PI		
Service interworking identifier	40	(Table B.1)
Document reference number	41	Activity Identifier
Checkpoint reference number	42	Serial Number
Reserved	43	
Acceptance of CDCL parameters	44	(Table B.1)
Storage capacity negotiation	45	(Table B.1)
Receiving ability jeopardized	46	User Data (in MIA SPDU)
Reserved	47	
Document type identifier	48	(Table B.1)
Reflect parameter values	49	Reflect Parameter Values
Reason (session and document)	50	Reason Code
	51	Calling Session Selector
	52	Called Session Selector/Responding Session Selector
	53	Second Resync Type
	54	Second Serial Number
	55	Second Initial Serial Number
	56	Upper Limit Serial Number
	57	Large Initial Serial Number
	58	Large Second Initial Serial Number
	59	
	60	Data Overflow
	61	
	62	
	63	
PGI		
Reserved for extension	64	(Table B.1)
Non-basic teletex terminal capabilities	65	(Table B.1)
	66	
	67	
	68	
	69	
	70	
	71	
PI		
Graphics character set	72	(Table B.1)
Control character set	73	(Table B.1)
Teletex page format	74	(Table B.1)
Miscellaneous teletex terminal capabilities	75	(Table B.1)
	76	
Number of dots for character box height	77	(Table B.1)
Number of dots for character box width	78	(Table B.1)
	79	
	.	
PGI	.	
	192	
	193	
	194	Extended User Data

Annex D

Differences between Protocol Version 1 and Protocol Version 2

(This annex does not form an integral part of this Recommendation | International Standard)

D.1 Protocol Version 1 of the session protocol is a subset of Protocol Version 2. Protocol Version 1 imposes restrictions on the length of the user data field, whereas Protocol Version 2 imposes no explicit restrictions on the length of the user data field.

The differences between the two protocol versions are indicated below:

a) the OVERFLOW ACCEPT SPDU and the CONNECT DATA OVERFLOW SPDU are not used when Protocol Version 1 has been proposed;

b) the maximum lengths of parameter values in SPDUs for each protocol version are listed in Table D.1;

c) when Protocol Version 1 has been selected, segmenting, as specified in 6.3.5, b) does not apply; only data SSDUs and typed data SSDUs may be segmented.

D.2 An implementation of the session protocol may limit the length of user data supported, based on the requirements of its SS-user or protocol version supported. Any such limitation is stated in the Protocol Implementation Conformance Statement. If no user of a session implementation requires more than 10k of user data during connection establishment, the implementation need not be able to send the CDO SPDU or receive the OA SPDU.

Implementations of Protocol Version 2 can interwork with implementations of Protocol Version 1 only by imposing a number of restrictions (all of which are valid, in terms of the conformance statement). These restrictions are:

a) User Data parameter value in the ABORT SPDU does not exceed 9 octets.

b) Reason Code parameter value in the REFUSE SPDU does not exceed 513 octets.

c) The User Data PGI unit is not present in GIVE TOKENS, GIVE TOKENS CONFIRM, ACTIVITY INTERRUPT, ACTIVITY INTERRUPT ACK, ACTIVITY DISCARD and ACTIVITY DISCARD ACK SPDUs. The User Data PGI unit in all other SPDUs does not exceed 512 octets.

d) Protocol Version 1 is proposed in the CONNECT SPDU. In this case the Extended User Data parameter and Data Overflow parameter in the CONNECT SPDU are not present.

NOTE – Protocol Version 2 may also be proposed, but to operate validly with a Protocol Version 1 only implementation, Protocol Version 1 should be selected.

As a consequence of Protocol Version 1 being selected:

e) segmenting, as specified in 6.3.5, b) does not apply. Only data SSDUs and typed data SSDUs may be segmented.

f) the OVERFLOW ACCEPT SPDU and CONNECT DATA OVERFLOW SPDU are not used;

NOTE – Implementations of the earlier edition of this Recommendation | International Standard which specified Protocol Version 1 can upgrade to and claim conformance with this edition of this Recommendation | International Standard by stating, in their Protocol Implementation Conformance Statement, the restrictions specified in a) to c) above; and by conforming to the specified procedure for rejecting SPDUs with "too much" user data (see A.4.3.1.2) (note that this requires that the implementation recognizes the Extended User Data parameter in the CONNECT SPDU and the Enclosure Item in the ABORT SPDU). This is a minimal implementation of Protocol Version 2 and will not satisfy the requirements of some ASEs.

Table D.1 – Maximum length of the User Data PV field for each protocol version

SPDU and parameter field	Maximum lengths of parameter values		Reference
	Protocol Version 1	Protocol Version 2	
CONNECT SPDU User data Extended user data	512 octets Not applicable	512 octets 10240 octets	8.3.1.20 8.3.1.21
CONNECT DATA OVERFLOW SPDU User data	Not applicable	65528 octets	8.3.3.4
ACCEPT SPDU Enclosure item User data	Not applicable 512 octets	1 octet (Note)	8.3.4.18 8.3.4.21
REFUSE SPDU Enclosure item Reason Code	Not applicable 513 octets	1 octet (Note)	8.3.5.9 8.3.5.10
FINISH SPDU Enclosure item User data	Not applicable 512 octets	1 octet (Note)	8.3.6.4 8.3.6.5
DISCONNECT SPDU Enclosure item User data	Not applicable 512 octets	1 octet (Note)	8.3.7.3 8.3.7.4
NOT FINISHED SPDU Enclosure item User data	Not applicable 512 octets	1 octet (Note)	8.3.8.3 8.3.8.4
ABORT SPDU Enclosure item User data	Not applicable 9 octets	1 octet (Note)	8.3.9.4 8.3.9.6
CAPABILITY DATA SPDU Enclosure item User data	Not applicable 512 octets	1 octet (Note)	8.3.14.3 8.3.14.4
CAPABILITY DATA ACK SPDU Enclosure item User data	Not applicable 512 octets	1 octet (Note)	8.3.15.3 8.3.15.4
GIVE TOKENS SPDU Enclosure item User data	Not applicable Not applicable	1 octet (Note)	8.3.16.4 8.3.16.5
PLEASE TOKENS SPDU Enclosure item User data	Not applicable 512 octets	1 octet (Note)	8.3.17.4 8.3.17.5
GIVE TOKENS CONFIRM SPDU Enclosure item User data	Not applicable Not applicable	1 octet (Note)	8.3.18.3 8.3.18.4
MINOR SYNC POINT SPDU Enclosure item User data	Not applicable 512 octets	1 octet (Note)	8.3.20.4 8.3.20.6
MINOR SYNC ACK SPDU Enclosure item User data	Not applicable 512 octets	1 octet (Note)	8.3.21.3 8.3.21.5
MAJOR SYNC POINT SPDU Enclosure item User data	Not applicable 512 octets	1 octet (Note)	8.3.22.4 8.3.22.6

Table D.1 *(concluded)* – Maximum length of the User Data PV field for each protocol version

SPDU and parameter field	Maximum lengths of parameter values		Reference
	Protocol Version 1	Protocol Version 2	
MAJOR SYNC ACK SPDU Enclosure item User data	Not applicable 512 octets	1 octet (Note)	8.3.23.3 8.3.23.6
RESYNCHRONIZE SPDU Enclosure item User data	Not applicable 512 octets	1 octet (Note)	8.3.24.3 8.3.24.9
RESYNCHRONIZE ACK SPDU Enclosure item User data	Not applicable 512 octets	1 octet (Note)	8.3.25.3 8.3.25.9
EXCEPTION REPORT SPDU Reflect parameter values	65531 octets	65531 octets	8.3.27.3
EXCEPTION DATA SPDU Enclosure item User data	Not applicable 512 octets	1 octet (Note)	8.3.28.3 8.3.28.5
ACTIVITY START SPDU Enclosure item User data	Not applicable 512 octets	1 octet (Note)	8.3.29.3 8.3.29.5
ACTIVITY RESUME SPDU Enclosure item User data	Not applicable 512 octets	1 octet (Note)	8.3.30.3 8.3.30.12
ACTIVITY INTERRUPT SPDU Enclosure item User data	Not applicable Not applicable	1 octet (Note)	8.3.31.3 8.3.31.5
ACTIVITY INTERRUPT ACK SPDU Enclosure item User data	Not applicable Not applicable	1 octet (Note)	8.3.32.3 8.3.32.4
ACTIVITY DISCARD SPDU Enclosure item User data	Not applicable Not applicable	1 octet (Note)	8.3.33.3 8.3.33.5
ACTIVITY DISCARD ACK Enclosure item User data	Not applicable Not applicable	1 octet (Note)	8.3.34.3 8.3.34.4
ACTIVITY END SPDU Enclosure item User data	Not applicable 512 octets	1 octet (Note)	8.3.35.3 8.3.35.5
ACTIVITY END ACK SPDU	Identical to MAJOR SYNC ACK SPDU		

NOTE – The length of the user data parameter is limited such that the total length (including SI and LI) of the SPDU does not exceed 65539 octets.

ICS 35.100.50

Descriptors: data processing, information interchange, network interconnection, open systems interconnection, data transmission, connection oriented transmission, session layer, communication procedure, protocols.

Price based on 168 pages